REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
10-03-2011	Final Technical Report	Aug 2008 - Mar 2011

4. TITLE AND SUBTITLE
Predicting Mobility using Statistics (PreMoStat)

5a. CONTRACT NUMBER
W912HZ-08-C-0059

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hackett, Douglas D.
Longoria, Raul G.
Solis, Javier

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Griffin Technologies
PO Box 276
Wynnewood, PA 19096

8. PERFORMING ORGANIZATION REPORT NUMBER
2011-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Corps of Engineers (USACE) Engineering Research and Development Center (ERDC)
3909 Halls Ferry Road, Vicksburg, MS 39180-6199

10. SPONSOR/MONITOR'S ACRONYM(S)
ERDC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution Statement A - Public distribution/availablility

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The purpose of this project was to develop methods and software to determine whether a given Small Unmanned Ground Vehicle (SUGV) can traverse a given terrain, when both the SUGV and the terrain are not known exactly. A simulation model of a real-world SUGV (iRobot PackBot) was developed in the ADAMS environment and used to simulate traversal of a variable-height step obstacle. For this project, a user subroutine was successfully integrated into the ADAMS model to predict deformable track-terrain interaction. A parameterizable UGV vehicle system model was implemented using the ADAMS command language. A key element of this model is a slip-sinkage model. This simulation model was validated using real-world data collected in a step validation fixture with sand and a variable-height curb.
In a related effort, this project developed methods for using UGV sensor data to estimate variables and parameters needed for traction force prediction. The methods were evaluated using data collected from experiments with a PackBot traversing various deformable and non-deformable surfaces.

15. SUBJECT TERMS
Unmanned Ground Vehicle, simulation, ADAMS, PackBot, deformable terrain, parameter estimation, simulation validation

16. SECURITY CLASSIFICATION OF: Unclassified			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Jody Priddy, ERDC GSL
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	NA	62	**19b. TELEPHONE NUMBER** *(include area code)* (601) 634-3015

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

PreMoStat

STTR A07-T026 Final Report

USACE ERDC, Vicksburg, MS

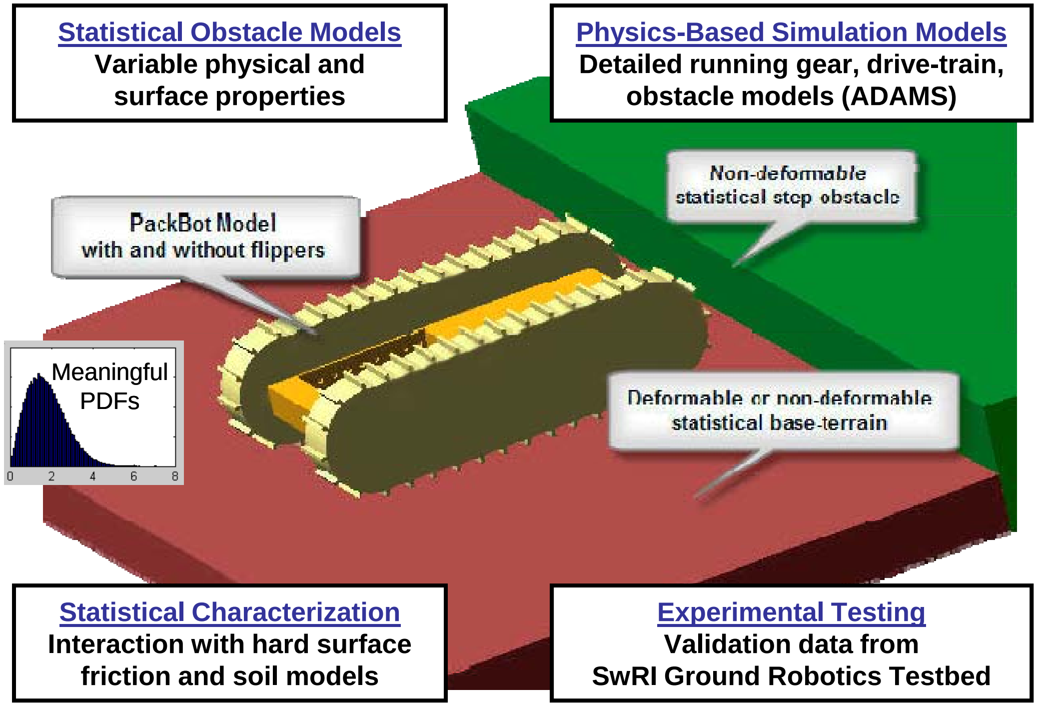
March 7, 2011

Griffin Technologies

University of Texas at Austin

Southwest Research Institute

PREdicting Mobility using STATistics (PreMoStat)



Phase I

- Develop Monte Carlo model of PackBot climbing step, varying step height and surface friction
- Validated predicted traverse with empirical test data

Phase II

- Extend methods to deformable terrain (e.g., sand)

Purpose:

- To develop methods and software to determine whether a given Small Unmanned Ground Vehicle (SUGV) can traverse a given terrain, when both the SUGV and the terrain are not known exactly:
  - Develop efficient UGV simulations that incorporate statistical variability in vehicle-terrain interactions
  - Establish experimental methods to validate statistical models for vehicle-terrain interactions for typical obstacles/terrains
  - Evaluate simulation efficacy using an Army-relevant SUGV on testbed terrains and obstacles

Results:

- A novel, efficient statistical simulation framework for predicting off-road robot mobility
- Quantification of prediction accuracy on realistic terrains and obstacles
- Validated models for vehicle-terrain interactions

Payoff:

- Increased survivability, reliability, and mission effectiveness in all terrain conditions
- Insight into observed robot performance in tests and in the field
- Model for integration into UGV simulation environments

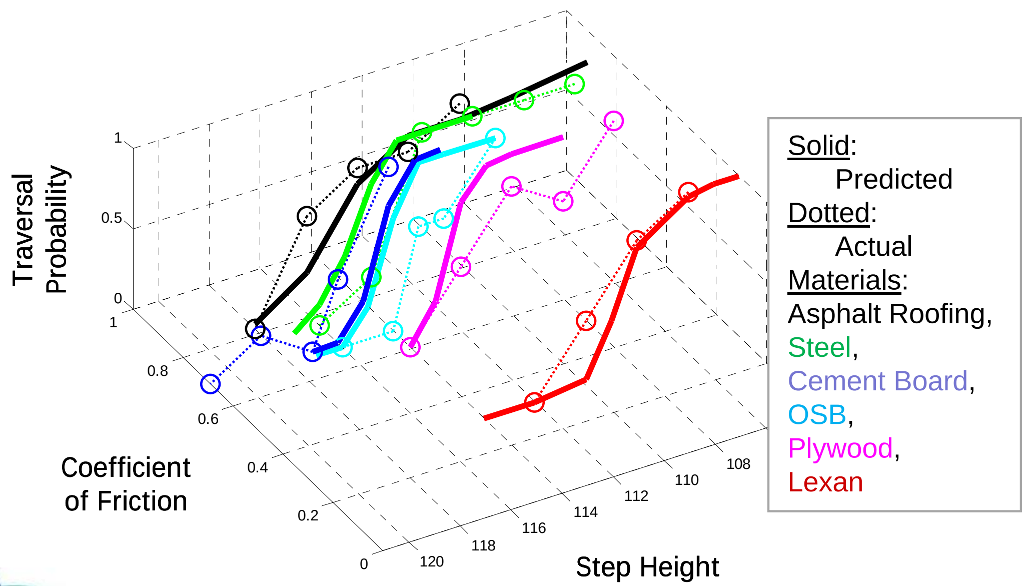
Phase 1 Summary

Collected real-world data on non-deformable steps

Built and validated a baseline mobility model

Found two key parameters: friction and step height

Confirmed that the model predicts the ability of a PackBot to climb a non-deformable step obstacles



Validation Data

Built a validation fixture at SwRI

- Platform motion using Vicon motion capture system
- PackBot internal data measurements
- On-board SwRI power logger
- Reference video

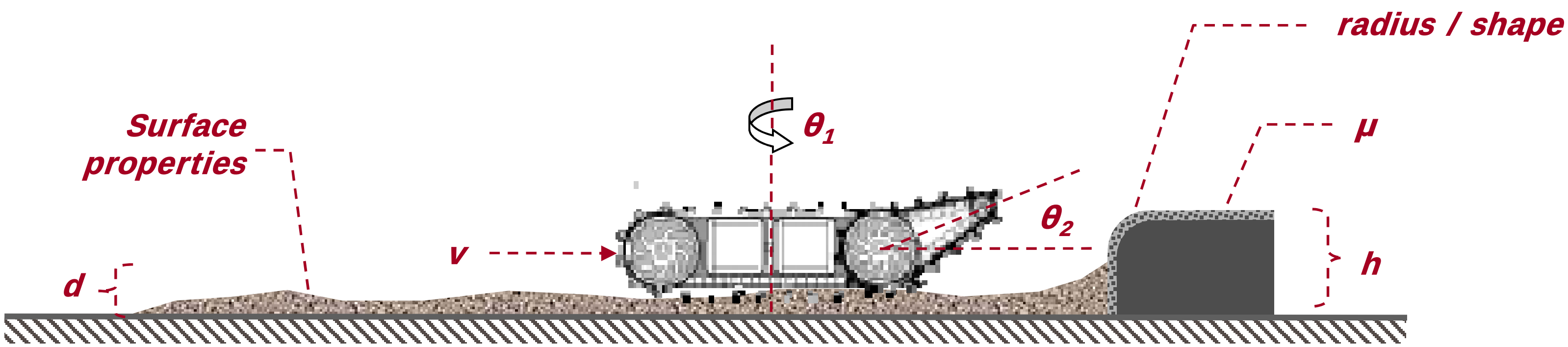
Collected non-deformable surface test data

- 5 Non-deformable surfaces, multiple heights, standard speed

Collected deformable surface validation data using sand

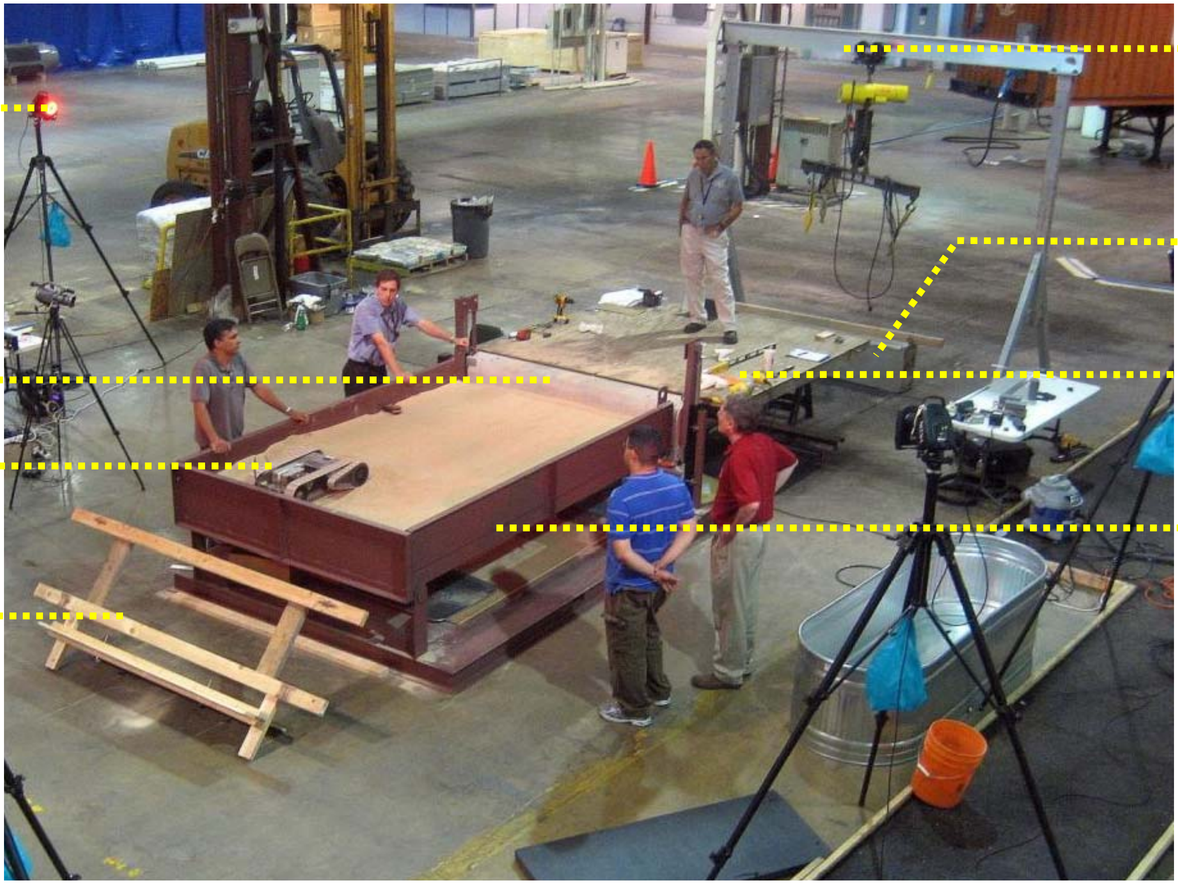
- 2 types of cement curbs, 4” and 1” radius
- Range of curb heights, from success to failure
- 4 speeds
- 2 sand depths
- 4 yaw angles

Fixture Parameters



- Curb height h
- Curb material μ
- Curb radius/shape r
- Yaw angle relative to curb θ_1
- Surface material depth d
- Velocity of the robot v
- Surface material properties (non-deformable)
- Flipper angle θ_2

The Fixture



Sand/Soil tray is fixed, curb and landing platform move vertically

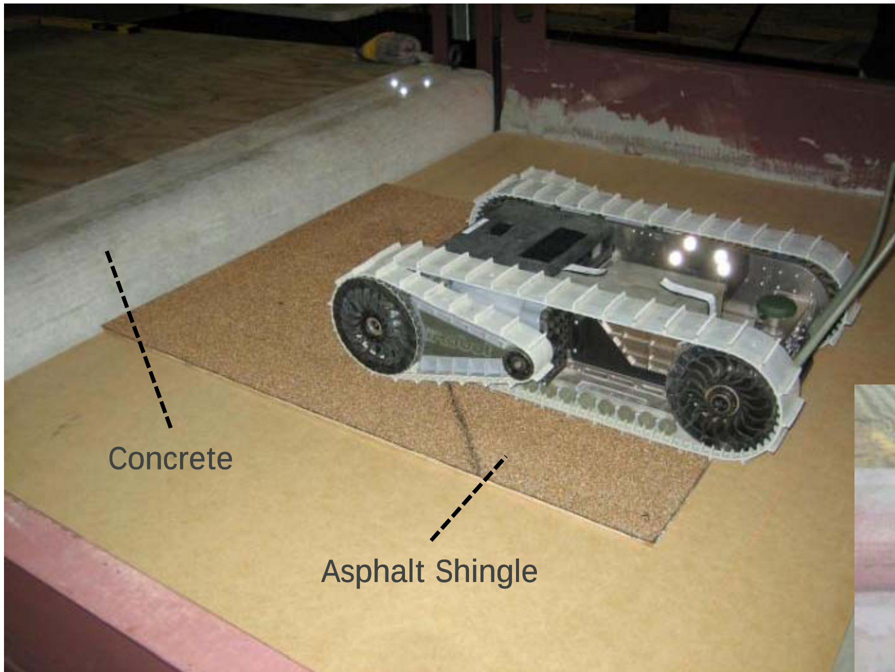
Fine Height Adjustment

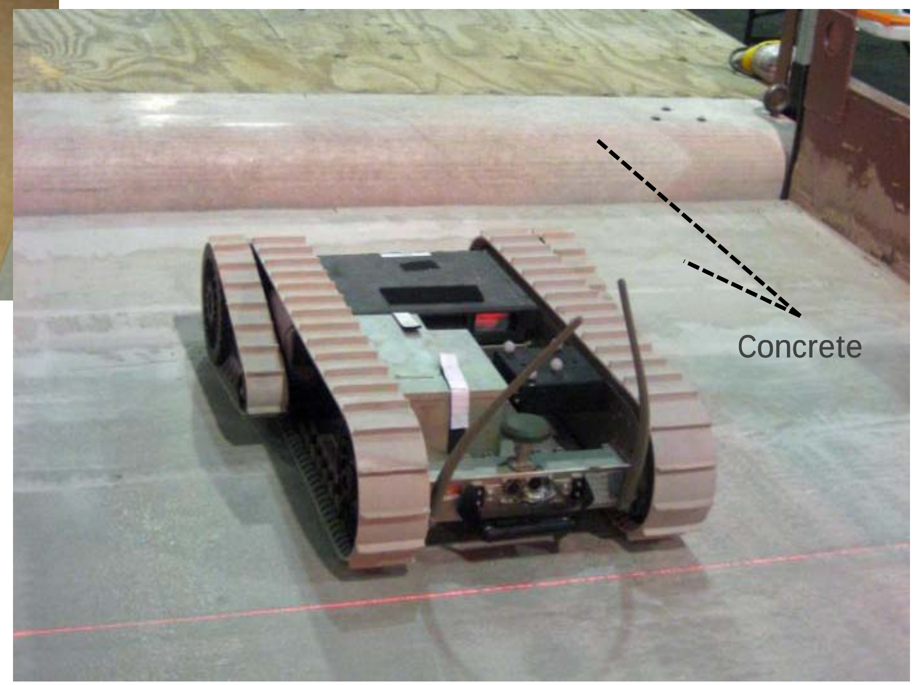
Adjustment mechanism raises/lowers curb 1mm per turn of the crank

Sand Depth and Leveling

Leveling device is adjusted to set the depth of sand

Non-Deformable Surfaces





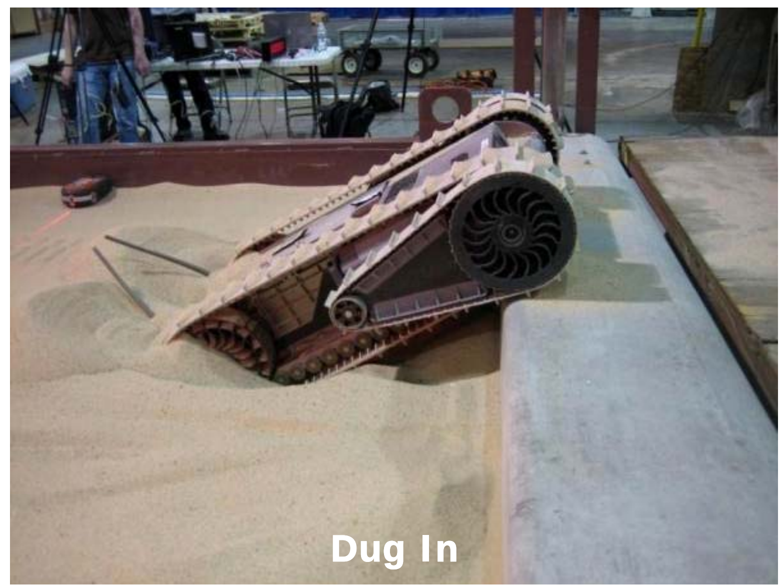
Testing the surfaces used in Phase 1 with 4” radius concrete curb

Some Failure Modes in Sand

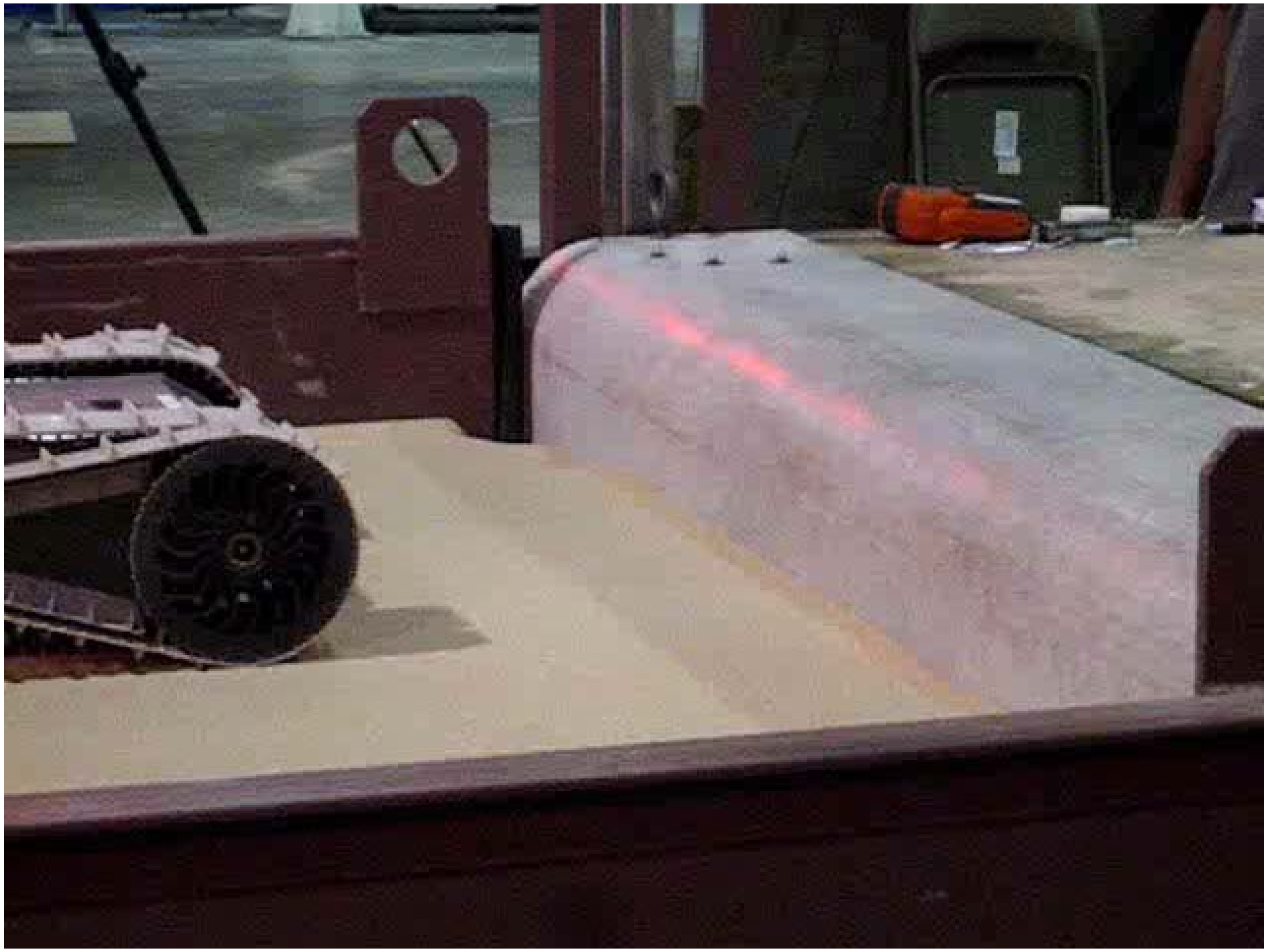

No Climb


Spun Out


High-Center


Dug In

Video



(Please click to play)

Disposition of Data

We used the data for validating the simulation model

A set of Matlab functions were developed to process and synchronize the data

The data was provided to performers on the DARPA Maximizing Mobility and Manipulation (M3) program

All of this data is available for further distribution

- Would need to look into any iRobot proprietary issues for release outside the Government

SwRI continues to refine their data collection processes and is available to collect further data

Developing a Simulation Model

Overview

1. What Was Done
2. Why Was This Done
3. Results
4. Comparison of Simulation and Test Results
5. Animations of Simulation Results
6. Development of Deformable Terrain Subroutine
7. Discussion of ADAMS Solver Simulation Process
8. Discussion on Slip-Sinkage Model
9. ADAMS Model Demo
10. Concluding Remarks

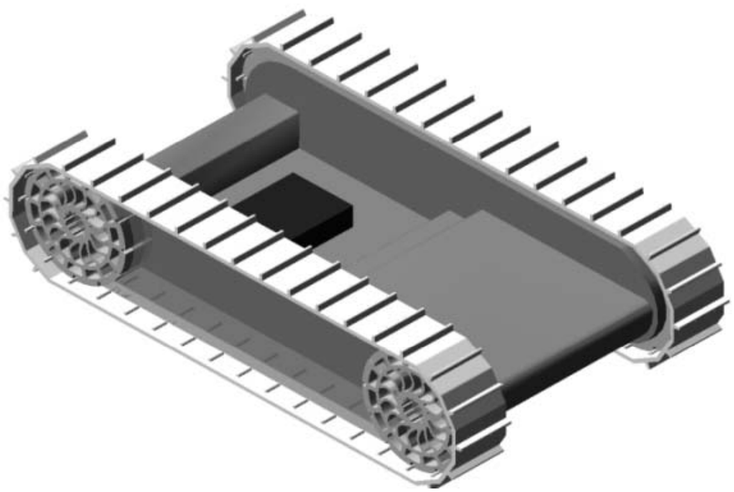
What Was Done

- Developed a subroutine for integrating specific track-terrain interaction models into ADAMS for predicting mobility on deformable terrains.

- Utilized ADAMS command language to efficiently build a parameterizable system model (rigid-bodies, forces, constraints, etc.)

- Developed a means for integrating slip-sinkage into the track-terrain interaction model.

- These new methods were fully implemented in ADAMS and evaluated using data collected during experiments with an iRobot PackBot at SwRI.



Why Was This Done

- Need a parameterizable track-terrain model for small scale UGV (PackBot) that could be used for statistical mobility prediction studies.

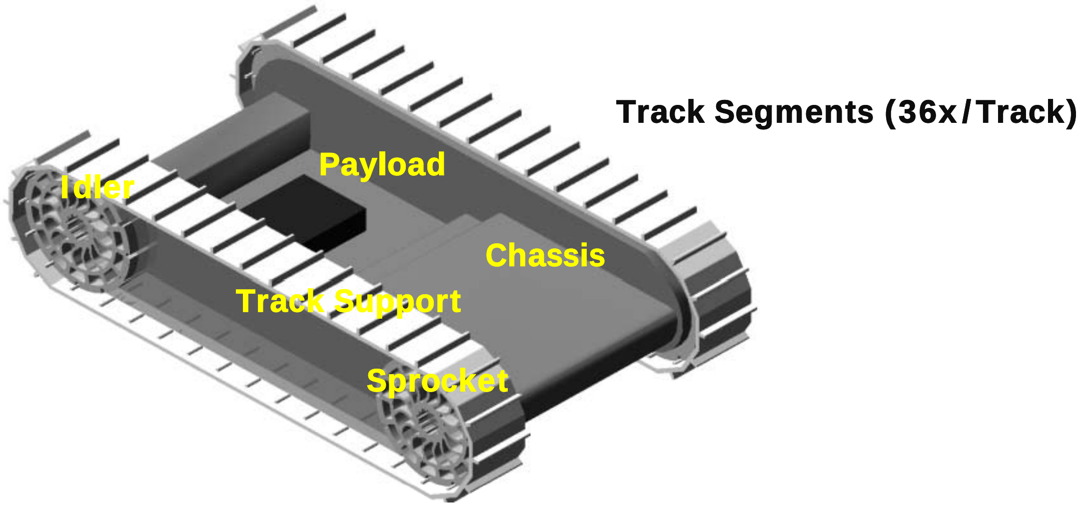
- Having a reliable parameterizable system model can have saving benefits (money and time) as opposed to physical testing.

- The ability to model highly deformable terrain interaction does not exist in ADAMS, so a customized subroutine was required.

- The ability to evaluate small UGVs operating in extreme maneuvers requires we be able to account for slip-sinkage.

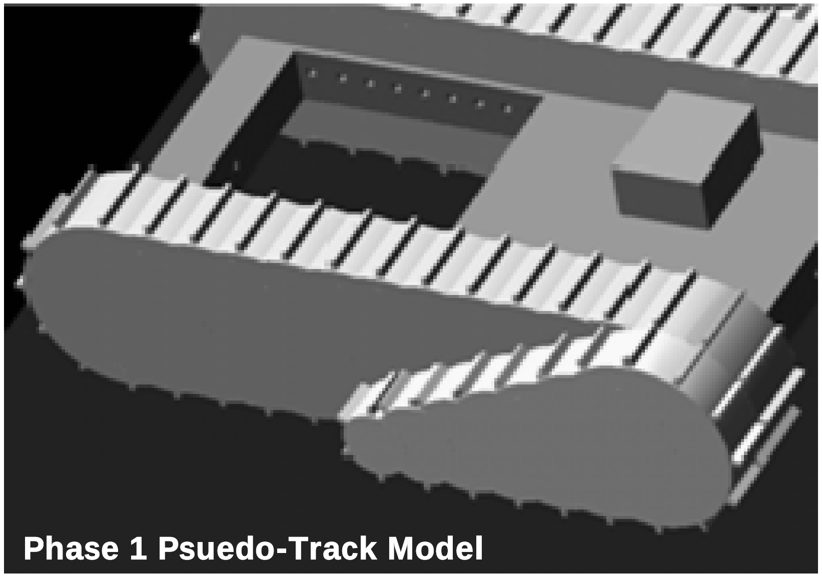
ADAMS Vehicle Model

- Chassis, Sprocket and Idler geometries imported from CAD system (SolidWorks).
- Track segment geometry was created in ADAMS environment allowing for parameterization.
- Solid model geometries can easily be changed by simply importing the modified geometry file.



Improved Dynamics Model

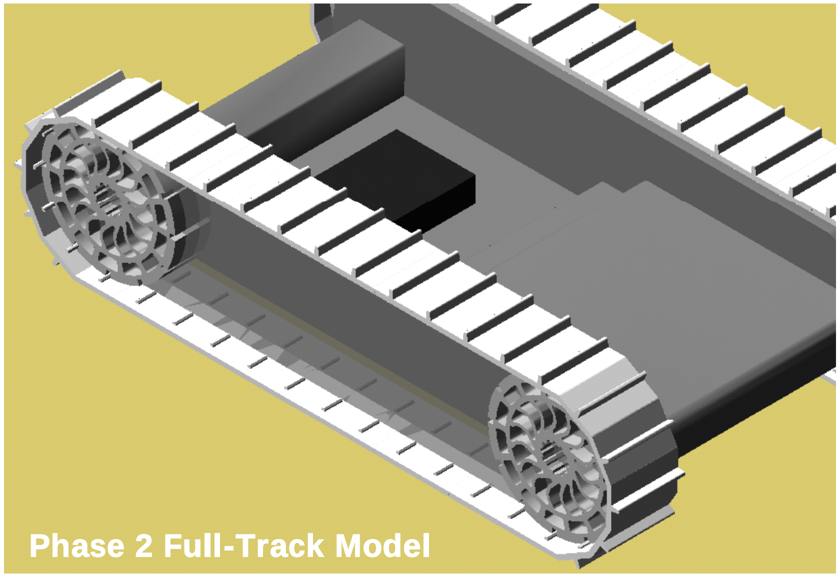
- Phase 1 PackBot psuedo-track model was a set of cascaded wheels
- Cleats engaging/disengaging with terrain introduced bouncing that reduced accuracy and increased simulation time
- Results show improved simulation performance even though new model has more bodies and constraints



Phase 1 Psuedo-Track Model

Pseudo-track

- *Cleats were attached to a set of cascaded wheels (Note curvature of track near cleat)*



Phase 2 Full-Track Model

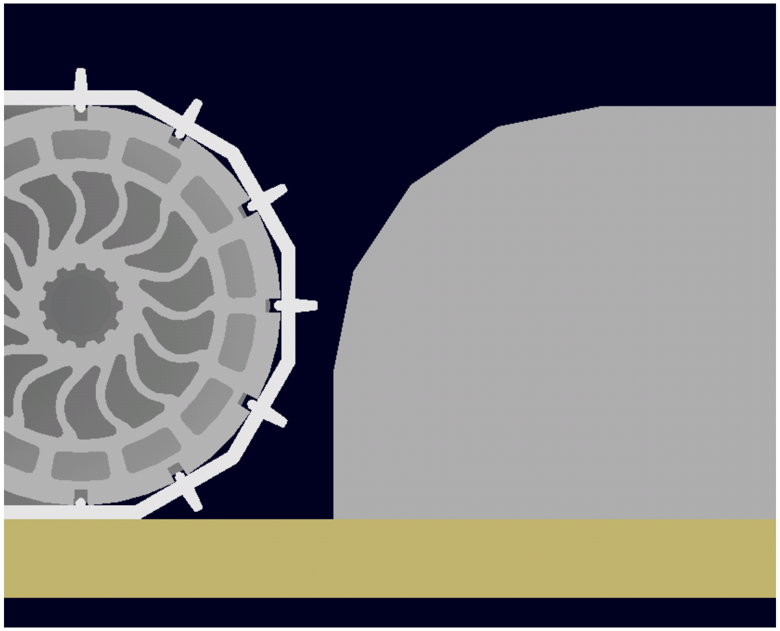
Segmented Track , 36 segments

- *Parameterized - cleat height, taper angle, etc.*
- *Spring/damper system used to constrain motion of track segments relative to one another*

Non-Deformable Terrain Model

PreMoStat

ADAMS 3-D contact model was implemented to compute both the normal and friction forces acting at the track segment-curb interface.



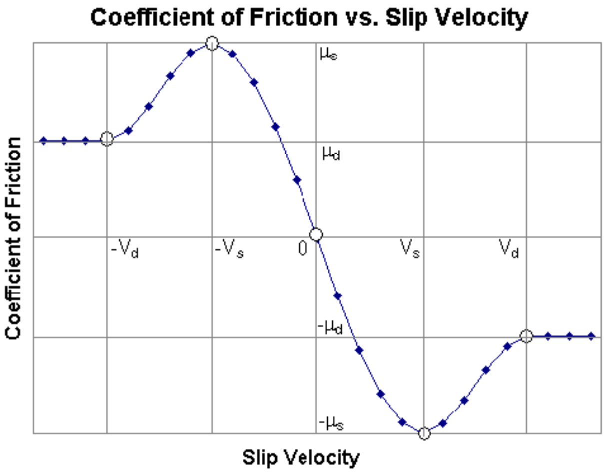
The normal force is computed using:

$$F_{normal} = k_s \Delta x^n + b_s f(\Delta x)\Delta\dot{x} \qquad \text{(Eq. 6)}$$

which is essentially a non-linear spring/damper system.

The friction force is modeled as

$$F_{friction} = \mu\left(v_{slip}\right) F_{normal} \qquad \text{(Eq. 7)}$$



where the relationship between the coefficient of friction and slip velocity is shown on the figure to the right.

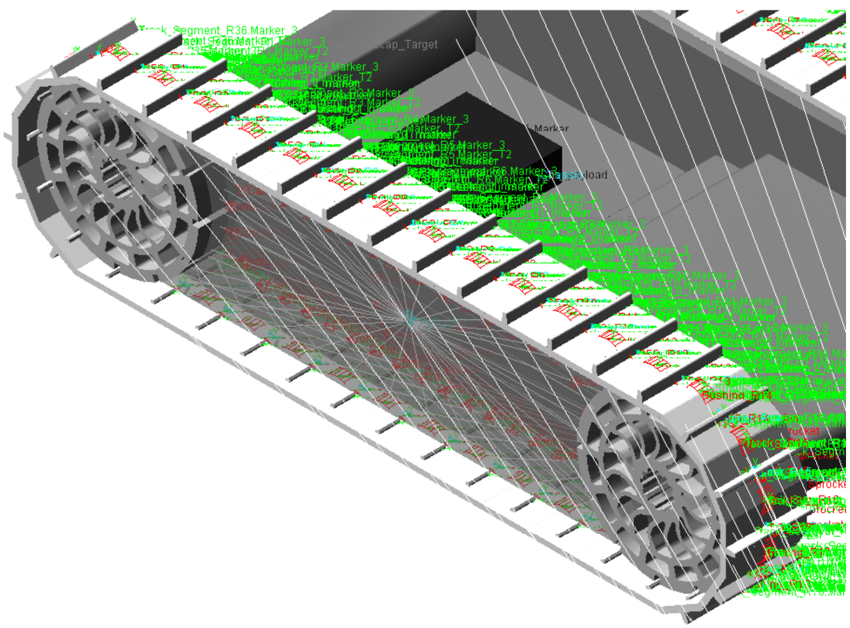
Δx = Penetration depth

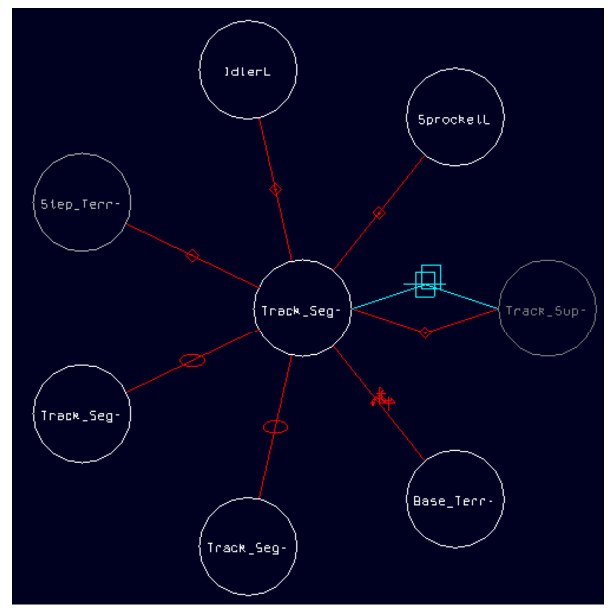
$\Delta\dot{x}$ = Penetration rate

ERDC GRIFFIN TECHNOLOGIES Robotics & Machine Perception

Track Model

- Discretized track belt (36x segments)
- 3-D contacts between segments, sprocket, idler and support
- Bushing elements constrain motion between joining segments
- In-plane constraint between segment and track support
- Parameterizable cleat geometry





ADAMS Graphical Topology Map

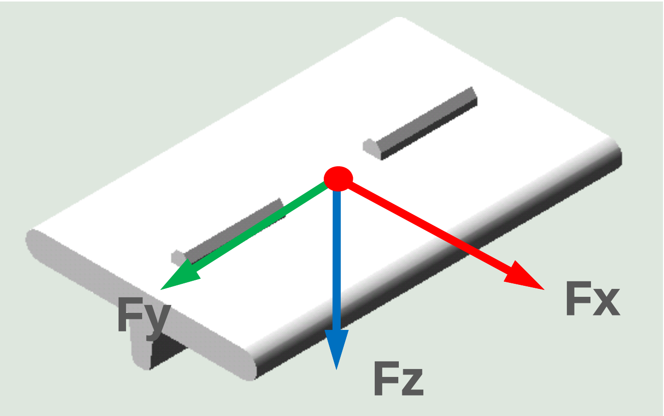
Deformable Terrain Model

Various deformable terrain models have been presented in the literature for predicting both the normal and shear loads acting at the track-terrain interface.

Two of the more popular methods were used for this study because of their ease of implementation and the fact that test methods exist to extract values of their empirical parameters. However, it is noted that PreMoStat model is not limited to these methods.

To start, a set of 3x orthogonal force elements are applied to each segment's center-of-mass (cm), which move with the body frame. For ease of explanation, we will assume an instance in time when the x, y, z force components are along the longitudinal, lateral and vertical directions, respectively.

The next sections discuss how both the normal and shear loads are computed and distributed among Fx, Fy, and Fz.



3x orthogonal force element acting a segment's cm.

Shear Displacement Model

PreMoStat

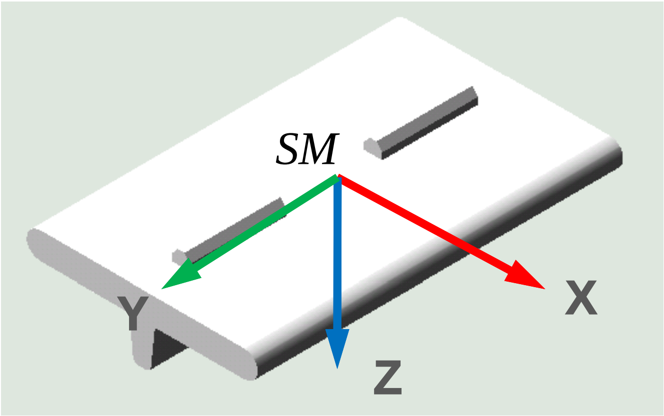
To generate traction force, shear displacement of the terrain must occur.

The total shear displacement for a single track segment is defined by:

$$j = \sqrt{j_x^2 + j_y^2}$$

where the components of the shear displacement are computed by integrating the slip velocity in the corresponding direction starting from initial contact with the terrain:

$$j_i = \int_0^{t_c} v_i dt \quad \text{for } i = x, y$$



j = absolute shear displacement

j_x = shear displacement along x-axis of SM

j_y = shear displacement along y-axis of SM

v_x = slip velocity expressed along x-axis of SM

v_y = slip velocity expressed along y-axis of SM

t_c = accumalitve time in contact with the terrain

SM = marker fixed to segment

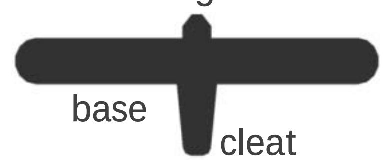
ERDC GRIFFIN TECHNOLOGIES Robotics & Machine Perception

Shear Force Model

The shear force model used for this analysis is based on a shear stress-shear displacement relationship proposed in [3] and is given by:

$$\tau = \left(c + \sigma \tan\phi\right)\left(1 - e^{-K/j}\right) \qquad \text{(Eq. 8)}$$

The total shear force acting on a given track segment is given by:



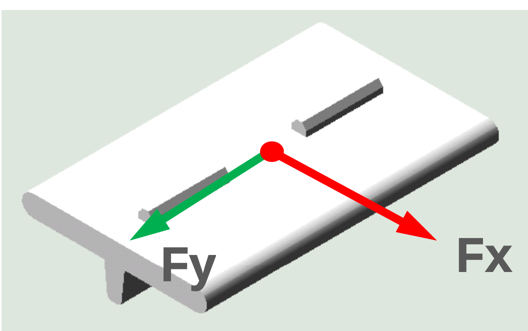
$$F_{shear} = \tau_{base} A_{base} + \tau_{cleat} A_{cleat} \qquad \text{(Eq. 9)}$$

Therefore, the shear force components are given by:

$$F_x = -F_{shear}\cos\theta$$
$$F_y = -F_{shear}\sin\theta \qquad \text{(Eq. 10)}$$

where,

$$\theta = \operatorname{atan2}\left(v_y / v_x\right)$$



c, ϕ, K = emperical parameters

σ = nomral pressure

v_x = slip velocity along x-axis

v_y = slip velocity along y-axis

A_{base} = base normal contact area

A_{cleat} = cleat noraml contact area

Normal Force Model

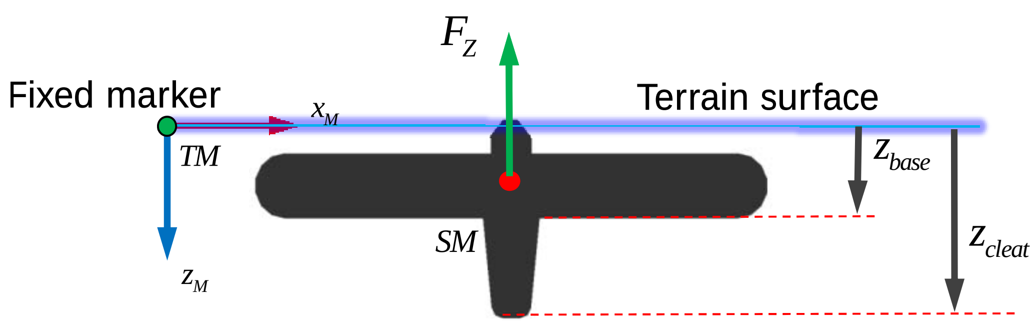
Therefore, the normal force acting on an individual track segment is computed as:

$$F_N = \left(K_{ss}^{-n} \cdot k \cdot z_{base} \cdot A_{base}\right) + \left(K_{ss}^{-n} \cdot k \cdot z_{cleat} \cdot A_{cleat}\right) + \left(b_{damp} \cdot \dot{z}_{base}\right) \qquad \text{(Eq. 11)}$$

where we have introduced a damping term to minimize oscillations allowing for better numerical stability and performance.

Therefore, the vertical force component is given by:

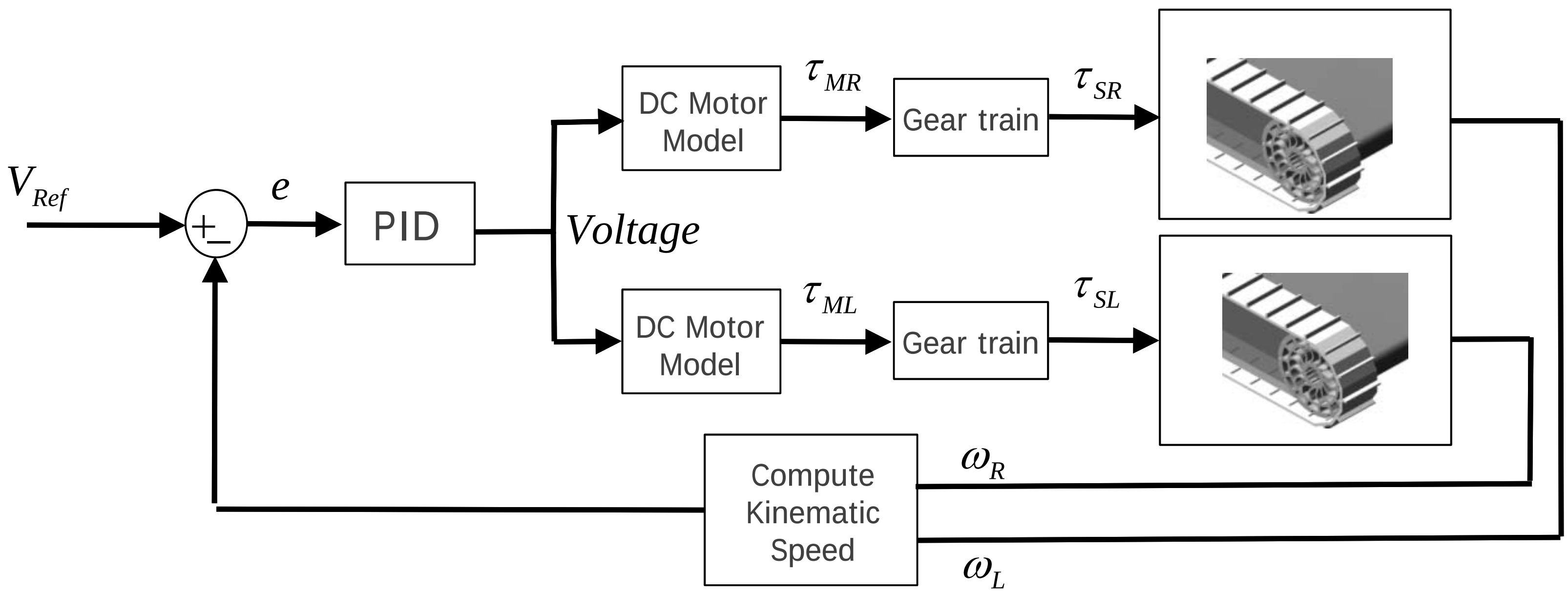
$$F_Z = -F_N \qquad \text{(Eq. 12)}$$



k = pressure sinkage parameter
z_{base} = static sinkage of base
z_{cleat} = static sinkage of cleat
A_{base} = area of base
A_{cleat} = area of cleat
$\dot{z}_{base}$ = sinkage rate of base
b_{damp} = damping parameter
TM = marker fixed to terrain
SM = marker fixed to segment

Control System

A simple PID controller was implemented to control the kinematic speed of the vehicle, which is similar to how the physical PackBot control system operates.



$\omega_{R,L}$ = right/left sprocket speed

$\tau_{MR,ML}$ = right/left motor torques

$\tau_{SR,SL}$ = right/left sprocket torques

R_W = sprocket hub radius

$$V_{Kin} = \frac{1}{2} R_W \left(\omega_R + \omega_L\right)$$

Results

- A user subroutine was successfully developed for predicting deformable track-terrain interaction and was implemented into an ADAMS UGV system model.

- A parameterizable UGV vehicle system model was successfully developed using the ADAMS command language allowing for a more automated building process.

- A slip-sinkage model was integrated into the deformable terrain model and shows promising results considering its simplicity in formulation.

- Because of the extensive detail of the track model, simulation of ~9 (sec) real time takes ~2hrs to complete.

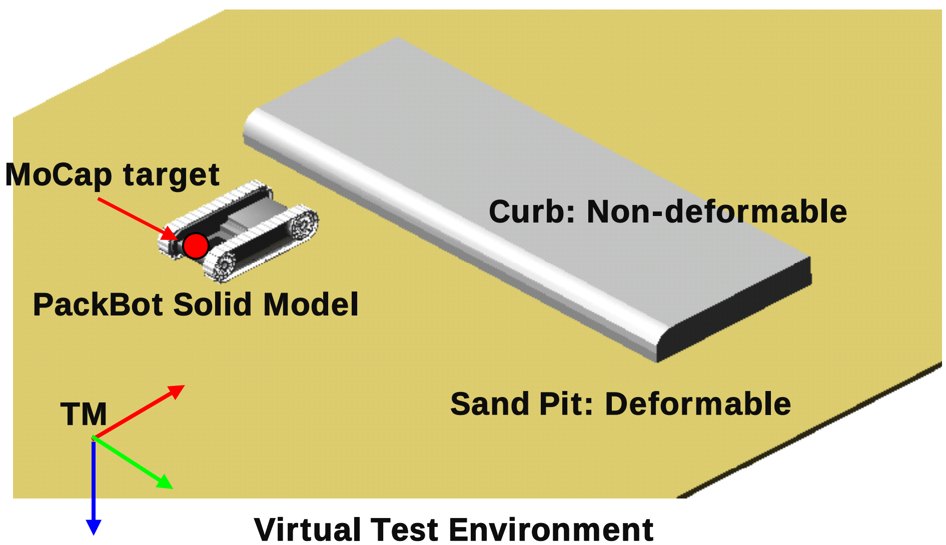
Simulation Test Case

To validate the ADAMS PackBot model, simulation results were compared to test data collected at SwRI.

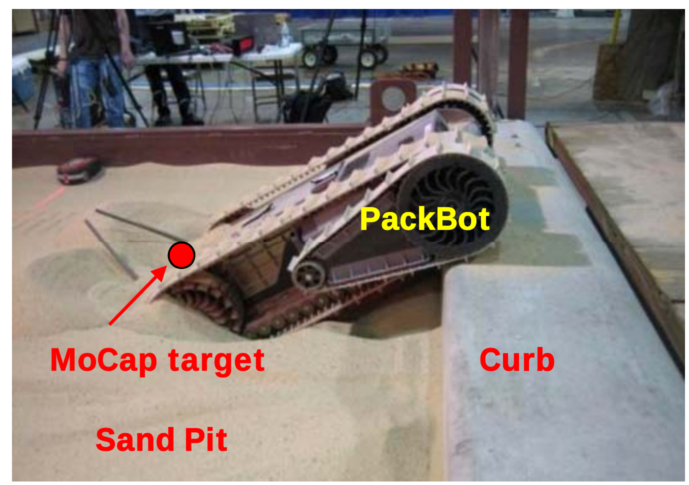
The test cases consists of the PackBot traversing over a curb obstacle set to various heights and radii with the base of the curb being a loose sandy terrain.

The speed of the PackBot was set to operate at different reference speeds, which was also accounted for in the simulation using a PID control scheme.

A motion capture system was used to measure the position/orientation and velocities of the PackBot.



Virtual Test Environment



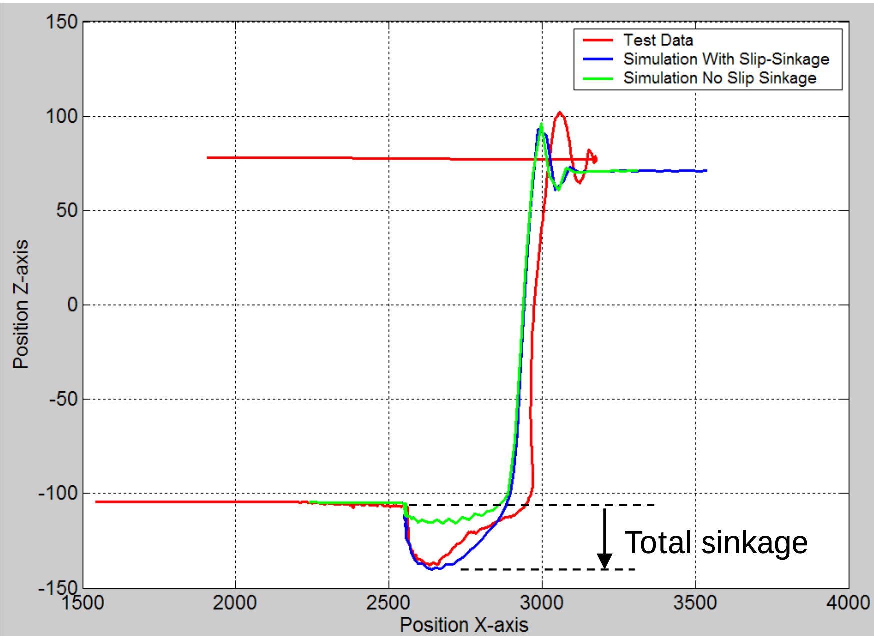
Physical Test Environment

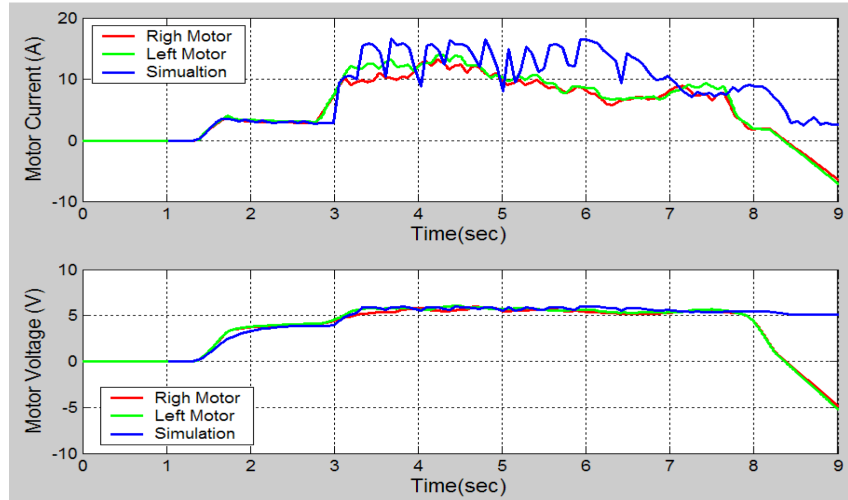
Simulation Results

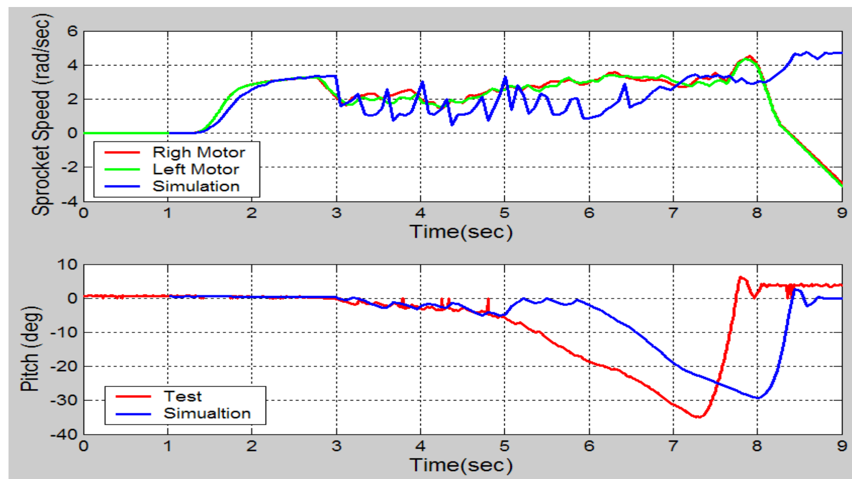
PreMoStat

The following test case was simulated and compared:

Curb Height = 163 (mm)
Curb Radius = 4 (in.)
Reference Speed = 300 (mm/sec)







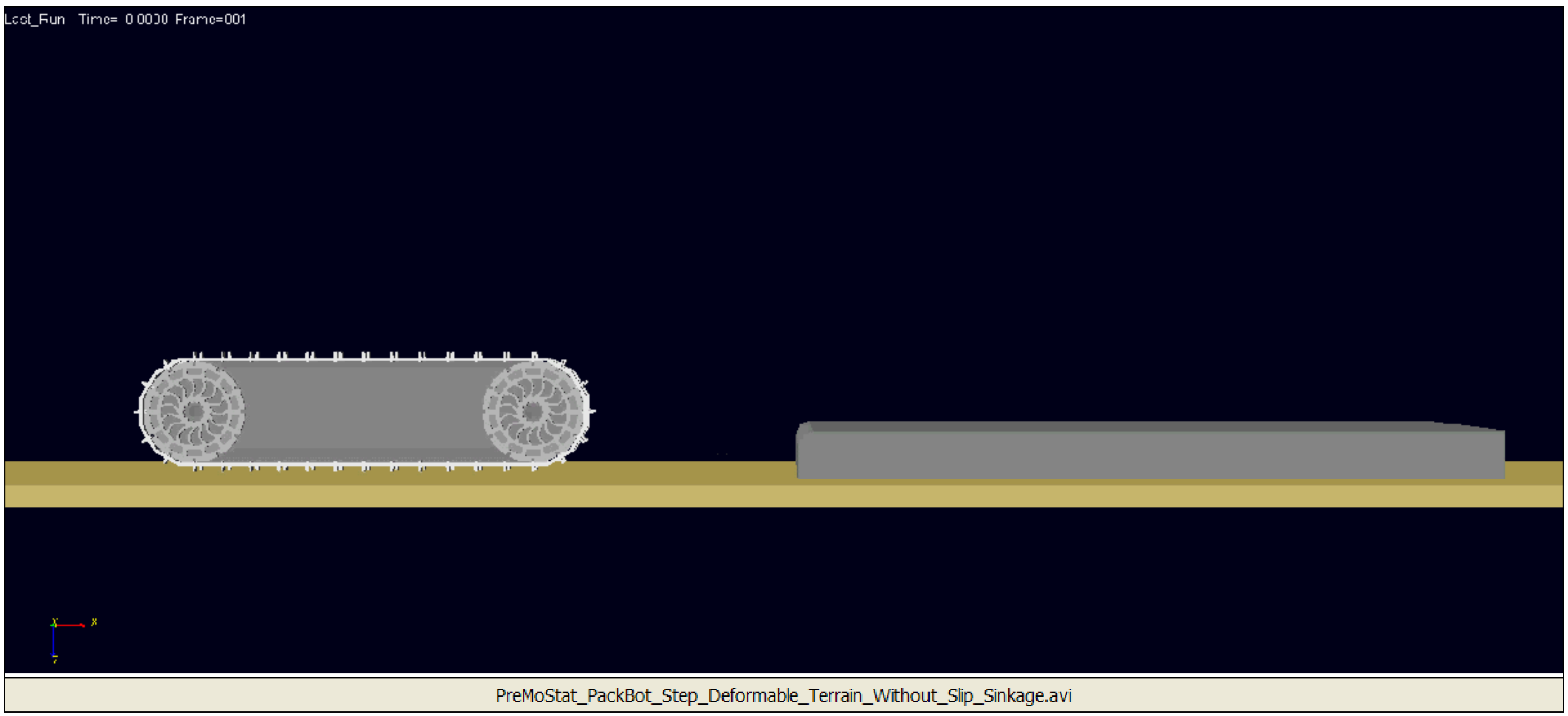
Max Static sinkage ~ 11.6 (mm) ~32 %
Max Slip-sinkage ~ 24.2 (mm) ~68 %
Max Total sinkage ~ 35.8 (mm)

ERDC GRIFFIN TECHNOLOGIES Robotics & Machine Perception

Animation With No Slip-Sinkage

This animation shows a simulation result of the ADAMS PackBot model traversing a curb of height 117 (mm) and radius of 1 (in.) at 300 (mm/sec).

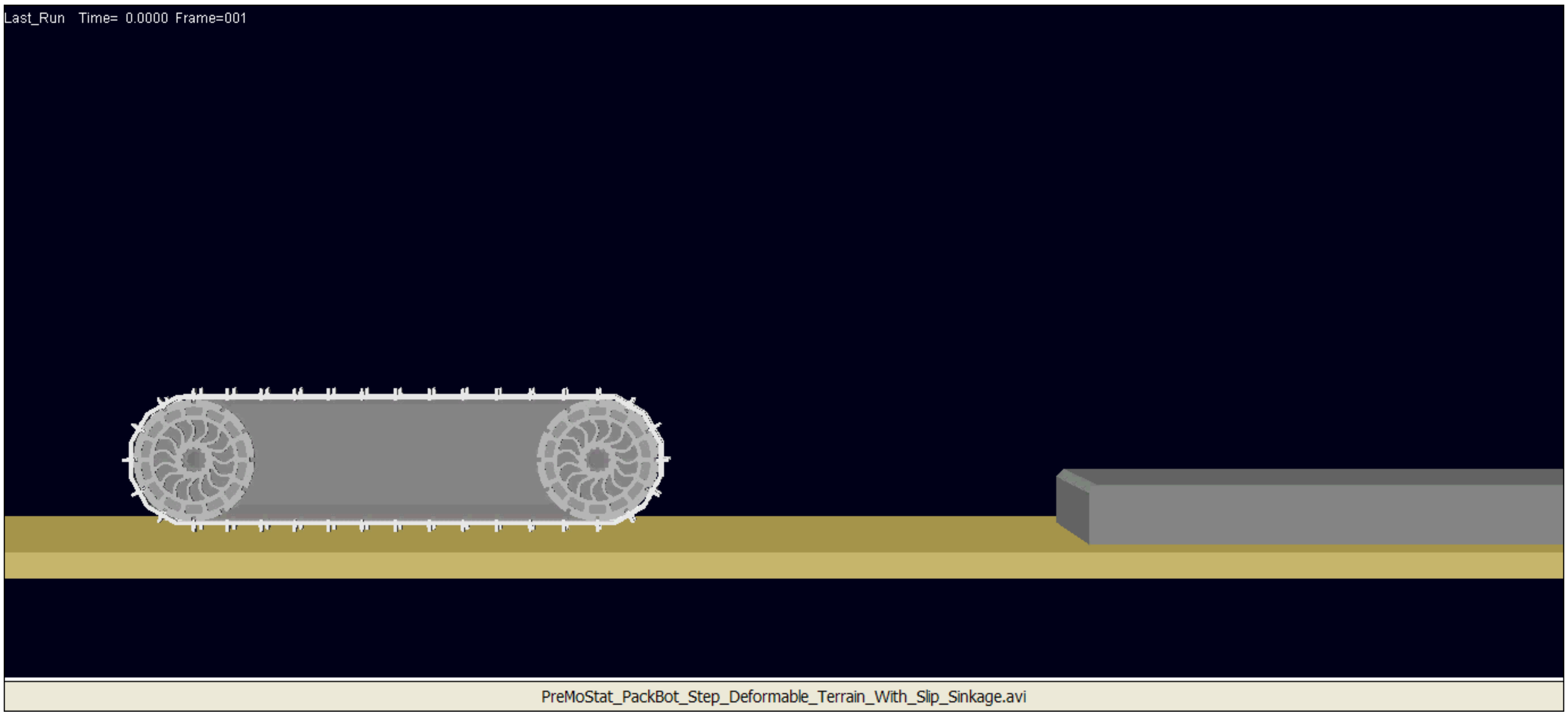
This version of the deformable terrain subroutine does not account for slip-sinkage, but only static sinkage.



Animation With Slip-Sinkage

This animation shows a simulation result of the ADAMS PackBot model traversing a curb of height 117 (mm) and radius of 1 (in.) at 300 (mm/sec).

This version of the deformable terrain subroutine accounts for slip-sinkage which is evident when compared to the previous animation.



Developing Terrain Subroutine

PreMoStat

ADAMS allows one to define customized force elements using user-defined subroutines coded in either Fortran or C++.

Because of the complex nature of deformable terrain interaction, including large terrain deformations, it was necessary for our work to develop a user-defined force subroutine.

To accomplish this task, we used a bottom-up process by breaking down the subroutine into multiple functions.

- Function → computing normal forces
- Function → compute shear forces
- Function → terrain query
- Function → error checking

Each of these functions were developed and tested on elementary models to simplify the debugging process.

Check_Params.f	2 KB	Fortran Source
Comp_Npressure_Bekker.f	2 KB	Fortran Source
Comp_Shear_Stress_Janosi.f	1 KB	Fortran Source
Find_Terrain_Element.f	3 KB	Fortran Source
PreMoStat_GFOSUB.f	14 KB	Fortran Source

PreMoStat_GFOSUB.f

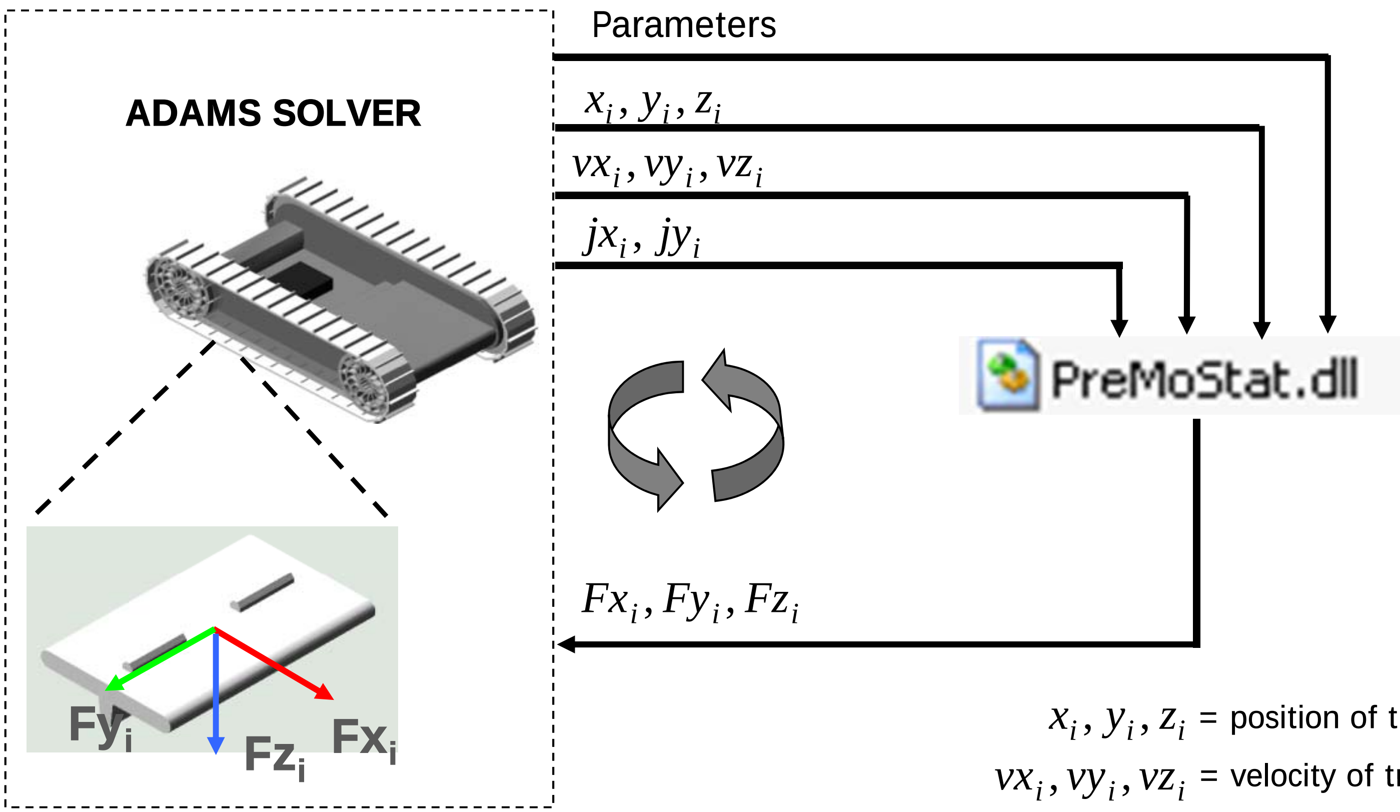
```
C ======================================================================
C FILE INFORMATION:
C   Author: Javier M. Solis
C   File: PreMoStat_GFOSUB
C   Created: 04-18-10
C   Mofified: 02-14-11
C ======================================================================
C DESCRIPTION:
C     PreMoStat subroutine used to model deformable terrains
C ======================================================================

      subroutine GFOSUB(id, time, par, npar, dflag, iflag, result)

      implicit none
C =============== Type and dimension statements ========================
C
C ---------------- External variable definitions -----------------------

      integer                            id
      double precision                   time
      double precision                   par( * )
      integer                            npar
      logical                            dflag
      logical                            iflag
      double precision                   result(6)

C   id            Identifier of calling GFORCE statement
C   time          Current time
C   par           Array containing passed parameters
C   npar          Number of passed parameters
C   dflag         Differencing flag
C   iflag         Initial pass flag
C   result        Array (dimension 6) of computed GFORCE
C                 components returned to ADAMS

C ---------------- Local variable and parameter definitions ------------

C *********************** Find_Terrain_Element.f ***********************
C       Arguements:
      integer          msize(2)
      double precision dxy(2), terrain_length, terrain_width
C       Output:
      integer          elem(2)

C ******************** Comp_Npressure_Bekker.f *************************
```

ERDC

GRIFFIN TECHNOLOGIES
Robotics & Machine Perception

ADAMS Solver Simulation Process

ADAMS Solver (numerical engine), passes the kinematic states of the i^{th} track segment to the PreMoStat.dll subroutine. In turn, the subroutine computes the normal and shear forces and passes this result back to ADAMS Solver. This process is repeated for each track segment at a given iteration time step.



x_i, y_i, z_i = position of track segment

vx_i, vy_i, vz_i = velocity of track segment

jx_i, jy_i = shear displacement of track segment

Fx_i, Fy_i, Fz_i = forces applied to track segment

Slip-Sinkage

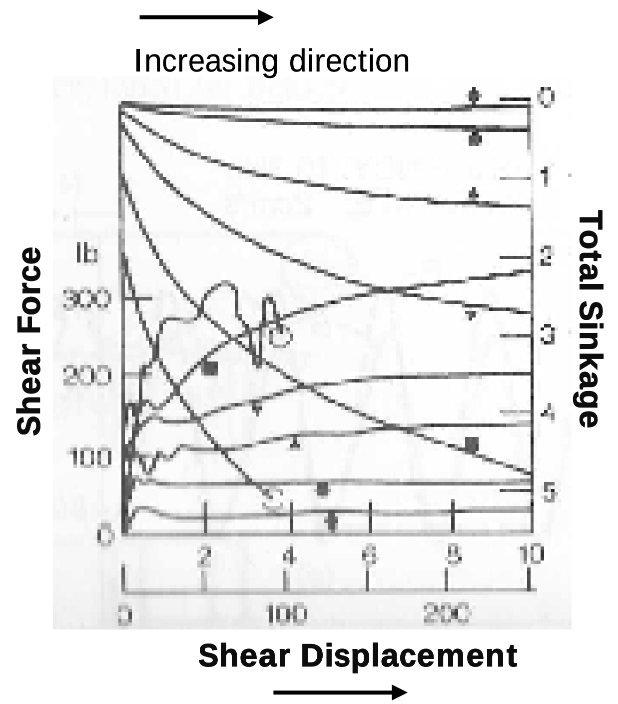
Shearing of the terrain leads to a phenomenon referred to as slip-sinkage.

As shearing increases, additional sinkage is introduced resulting in bulldozing.

The total sinkage of the vehicle is the sum of the static sinkage and the additional sinkage due to shearing:

$$z_{total} = z_o + z_j \qquad \text{(Eq. 1)}$$

The figure to the right shows how the total sinkage varies with both shear displacement and normal pressure. The difficulty of implementing slip-sinkage is separating the contributions of the two components.



Experience has shown that for small UGVs (like PackBot), the component of total sinkage due to slip is dominant.

Testing has also shown that slip-sinkage can significantly effect the mobility of small UGVs under extreme maneuvers such as step climbing and zero radius turns.

Slip-Sinkage Model

In [1], Lyasko used a conservation of energy approach and proposed that the total sinkage of a tracked vehicle can be captured using:

$$z = K_{ss} z_o \qquad \text{(Eq. 2)}$$

where

$$K_{ss} = \frac{1+i}{1-0.5i} \qquad \text{(Eq. 3)}$$

The significance of this approach is that the total sinkage is expressed as a function of slip (or shear displacement) and static sinkage, both of which can be computed directly. Another advantage of this approach is that no additional empirical parameters have been introduced.

z_o = static sinkage

z = total sinkage (static + sinkage due to slip)

i = slip, which can expressed as (j/x) for straight-line motion

j = shear displacement

x = position of track segment relative to track

Slip-Sinkage Model (Cont.)

The normal pressure acting on each track segment is based on a pressure sinkage relationship proposed in [2] and is given by:

$$p = \left(\frac{k_c}{b} + k_\phi \right) z_o^{\ n} = k \cdot z_o^{\ n} \qquad \text{(Eq. 4)}$$

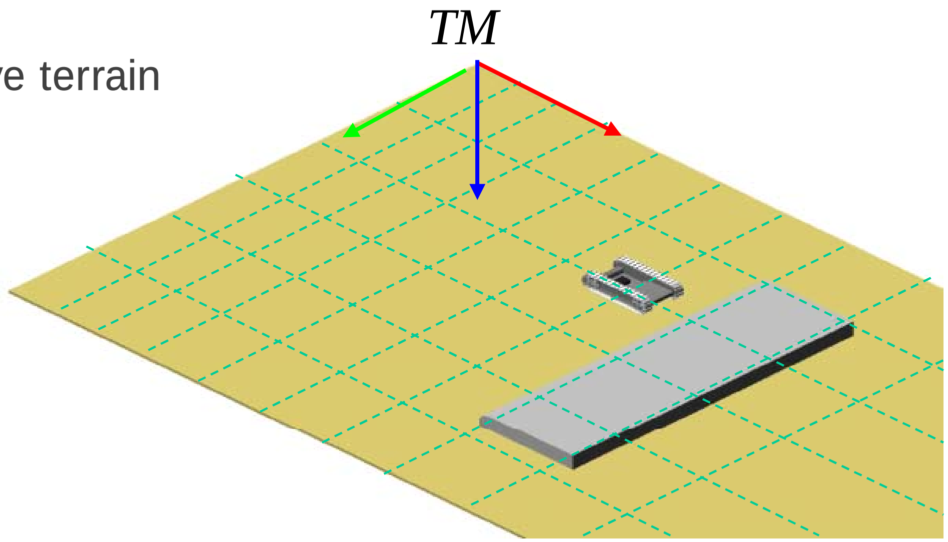
Accounting for slip-sinkage and inserting (Eq. 2) into (Eq. 4) and rearranging yields:

$$\boxed{p = K_{ss}^{-n} \cdot k \cdot z^n} \qquad \text{(Eq. 5)}$$

Notice as slip (or shear displacement) increases, the value of K_{ss}^{-n} decreases reducing the stiffness k, which in turn yields additional sinkage.

To account for terrain "memory", a primitive terrain query model was implemented.

The value of K_{ss}^{-n} for each element of the discretized terrain is "tracked" over the simulation.



Conclusions

- The PreMoStat UGV (PackBot) model is in a state to where it is ready to be extensively compared with test results, including investigating statistical variability.

- However, to improve the accuracy of the UGV (PackBot) system model, the values of certain parameters need to be measured/estimated, namely:
    - Soil parameters related the shear stress
    - Parameters related to damping and stiction through the drivetrain
    - Control parameters

- Although the effects of slip-sinkage have been accounted, the associated losses (bulldozing) should be integrated to more accurately predict mobility performance, especially for small radius turn maneuvers.

References

[1] Lyasko, M. Slip sinkage effect in soil-vehicle mechanics. J Terramechanics. 2010;47:21-31.

[2] M.G. Bekker. Introduction to Terrain-Vehicle Systems. Ann Arbor, MI: University of Michigan Press, 1969.

Vehicle-Based Track-Terrain Parameter Estimation

The Project

We developed methods for using UGV sensor data to estimate variables and parameters needed for traction force prediction.

The methods were evaluated using data collected from experiments with an iRobot PackBot traversing various deformable and non-deformable surfaces.

Data	K, m	Cohesion, Kpa	Phi, degree	Fmax, N
Obsvn 1	0.0012	740	15.697	43.789
Obsvn 2	0.0015	735	16.075	44.901
Obsvn 3	0.0014	750	16.415	45.905

Why was this done

We lack validated instruments for determining terrain parameters that can be used to predict small UGV mobility.

Need a way to collect data for guiding nominal parameter settings for vehicle-terrain interaction in UGV simulation studies.

There appear to be no databases available that can provide insight into the type of statistical variability we might expect to see in the parameters needed for models commonly used in vehicle mobility predictions.

Results and Conclusions

PreMoStat

We showed that an instrumented robotic vehicle can be used to estimate time-varying track slip and vehicle slip angle, variables critical in determining:

a) longitudinal and lateral friction coefficients in a baseline skid-steer model, and

b) cohesion, friction angle, and deformation modulus for model-based traction force prediction.

These preliminary results support related efforts reported in the literature, and suggest that SUGVs with onboard sensors may provide a means for building a mobility database for small-scale vehicle platforms.

ERDC GRIFFIN TECHNOLOGIES Robotics & Machine Perception

Recommendations

Additional testing is needed to assess accuracy against results from standard instruments, if available, as well as to generate a preliminary database useful for statistical

These 'sensor-endowed' vehicles are being developed and deployed on diverse terrains, and there could be a way to use this data for the benefit of prediction and design.

Support continued efforts to enable development of a mobility database using vehicle-based 'data mining'.

Documentation

T.M. Dar, R.G. Longoria, "Slip Estimation for Small-Scale Robotic Tracked Vehicles," 2010 American Control Conference, Baltimore, MD, June 30-July 02, 2010.

T.M. Dar, R.G. Longoria, "Estimating Traction Coefficients of Friction for Small-Scale Robotic Tracked Vehicles," 2010 Dynamic Systems and Control Conference, Cambridge, MA, Sept 13-15, 2010.

T.M. Dar, "Vehicle-Terrain Parameter Estimation for Small-Scale Robotic Tracked Vehicles," Doctoral Dissertation, Department of Mechanical Engineering, The University of Texas at Austin, December 2010.

In progress: journal article for J. of Terramechanics + another to J. of Dyn. Sys. Measurement and Control (ASME)

Methodology Used

The approach developed combines the use of Extended Kalman Filters (EKF) and Generalized Newton Raphson (GNR) methods in a multi-tiered algorithm.

Implicit in this approach are model bases that approximate the vehicle dynamics and the vehicle-terrain interaction.

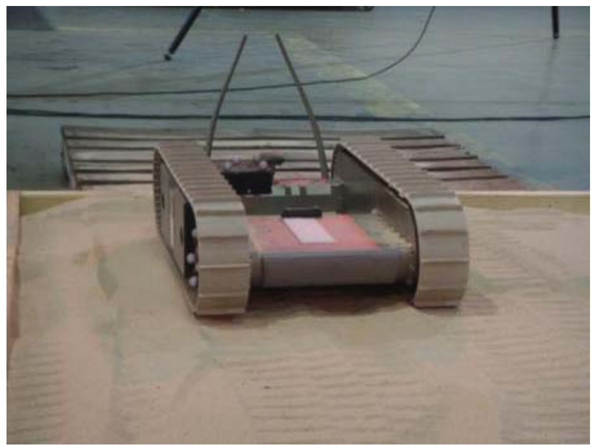
Experiments were designed accordingly:

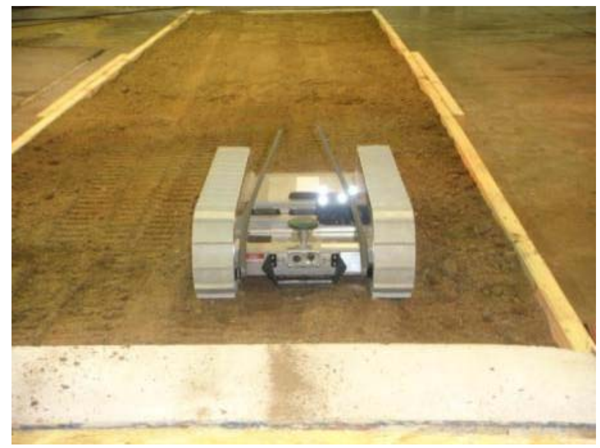
1. Ad hoc U-turns during sand pit testing

2. Field and indoor testing on various terrains

3. Prepared straight-line tests on sand and soil + step

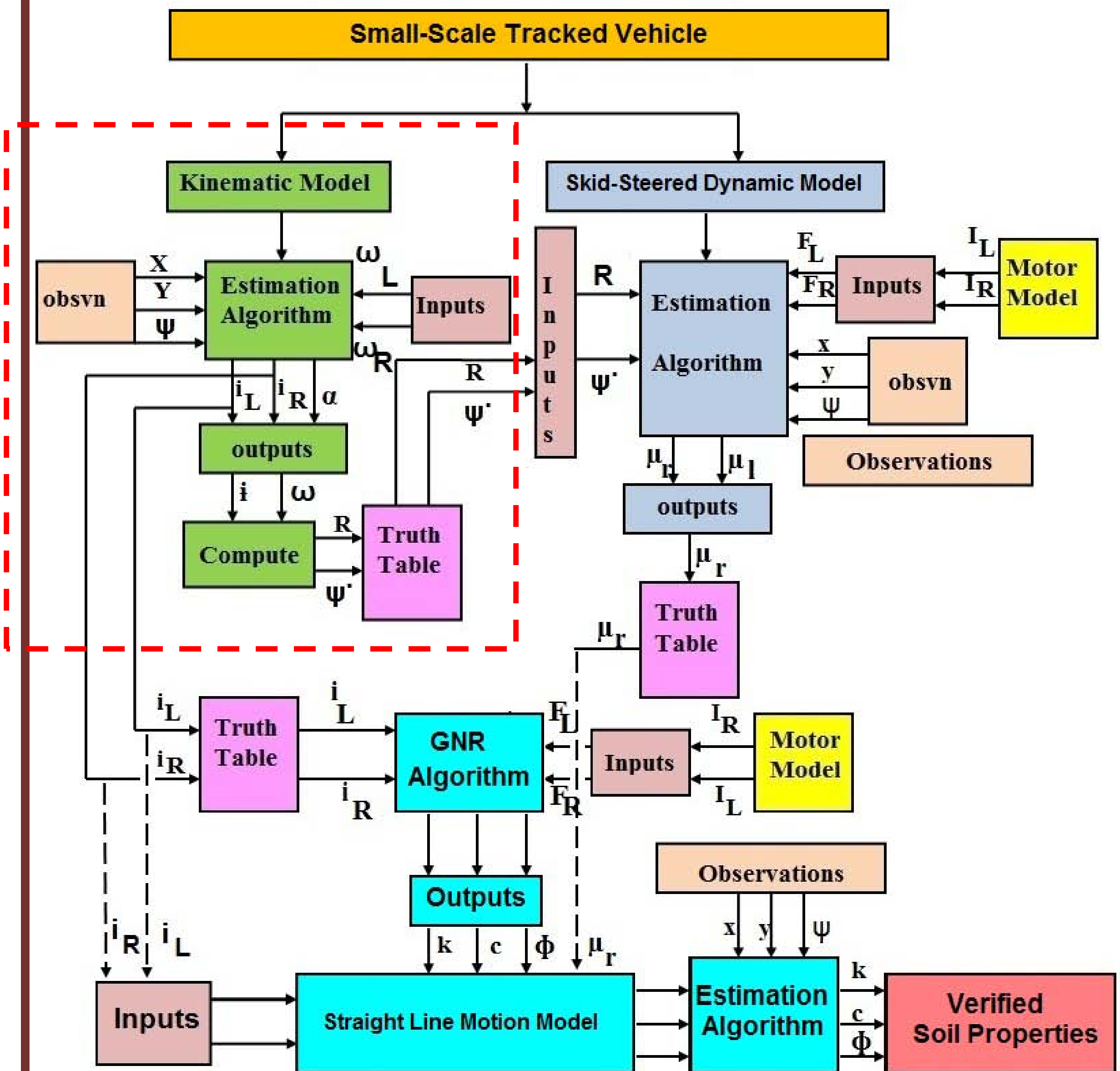
PreMoStat





Indoor tests employed the Vicon MoCap to track vehicle motion, while outdoor tests used a differential GPS.

Algorithm – 1



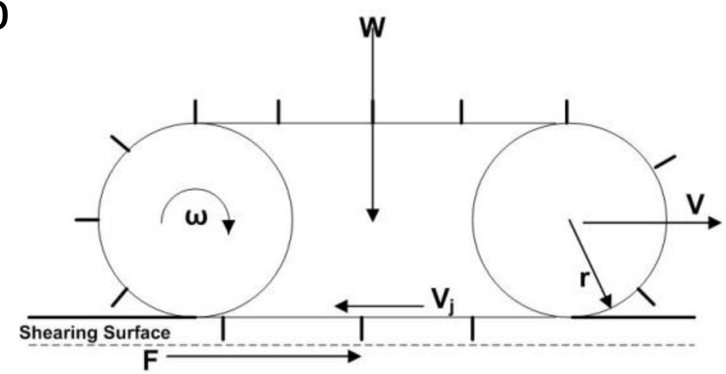
- α = slip angle
- c = soil cohesion
- F_L = tractive force left track
- F_R = tractive force right track
- i = slip on left/right track
- i_L = slip on left track
- i_R = slip on right track
- k = deformation parameter
- obsvn = observations
- ϕ = soil friction angle
- ψ = yaw angle
- $\dot{\psi}$ = yaw rate/turning rate
- I_L = current left motor
- I_R = current right motor
- ω_L = angular speed left sprocket
- ω_R = angular speed right sprocket
- μ_l = lateral coefficient of friction
- μ_r = longitudinal coefficient of friction
- x = displacement in x-direction body fixed coordinates
- y = displacement in y-direction body fixed coordinates
- X = displacement in X-direction inertial coordinate
- Y = displacement in Y-direction inertial coordinate

Slip Estimation

Having a good measure of slip is critical to traction force parameter estimation.

Each track has slip,

$$i = 1 - \frac{V}{r\omega} = 1 - \frac{V}{V_t} = \frac{V_t - V}{V_t} = \frac{V_j}{V_t}$$



Track slip impacts shear displacement along the track-terrain interface.

Our process: directly calculated slip using onboard measurements of encoder speeds together with approximated speeds based on position measurements made using either MoCap (indoor) or DGPS (outdoor).

This data, along with PackBot onboard data, was passed to algorithm.

Model for Slip Estimation

PreMoStat

$$x(k)=\begin{bmatrix} X(k) \\ Y(k) \\ \Psi(k) \\ i_L(k) \\ i_R(k) \\ \alpha(k) \end{bmatrix}$$

From an ideal kinematic steer model for a tracked vehicle, introduced slip in track and side slip, so full (kinematic) state equations become:

$$\dot{x}(k)=\begin{bmatrix} \frac{r}{2}[\omega_L(1-i_L)+\omega_R(1-i_R)]\left(\cos\psi+\tan\alpha\sin\psi\right) \\ \frac{r}{2}[\omega_L(1-i_L)+\omega_R(1-i_R)]\left(\sin\psi-\tan\alpha\cos\psi\right) \\ \frac{r}{2B}[\omega_R(1-i_R)-\omega_L(1-i_L)] \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

*Assume slip and side slip dynamics are zero, lacking dynamic model. So we need SNC to compensate for this model uncertainty.

GRIFFIN TECHNOLOGIES Robotics & Machine Perception

Estimation Algorithm – EKF

Observation model:

$$Z_k = [X_k; Y_k; \Psi_k]$$

$$z_k = Z_k - G(X_k^*, t_k)$$

$$\tilde{H} = \partial G(X_k^*, t_k) / \partial X_k \qquad \tilde{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The model basis along with the EKF construction enables us to come up with 'best estimates' of slip on each track and vehicle slip angle.

Further, we can embed parameters as 'states to be estimated' and all are computed in recursive manner

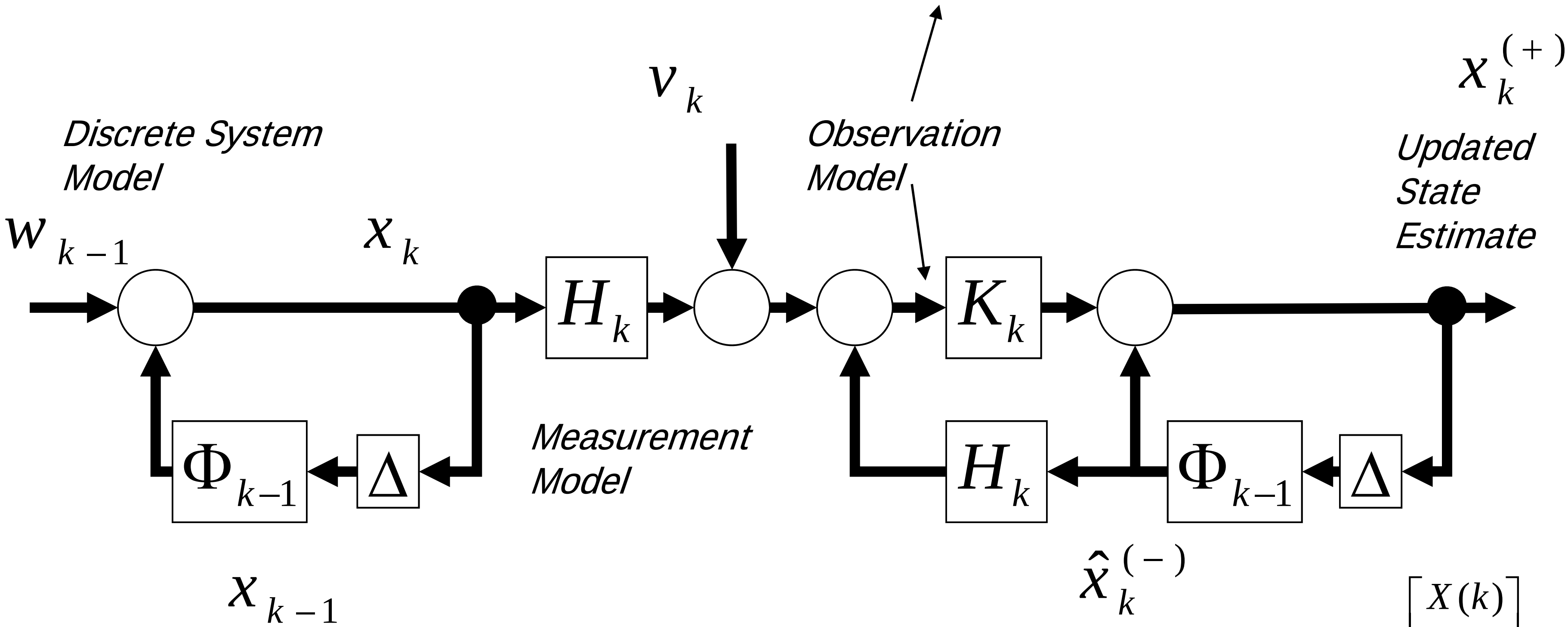
Kinematic model solved each step

$$\dot{X} = F\left(X^*, t\right) \Rightarrow X^*\left(t_{k-1}\right) = \hat{X}_{k-1}$$

$$\dot{\phi}(t, t_{k-1}) = A(t)\phi(t, t_{k-1}) \,\forall, \quad \phi(t_{k-1}, t_{k-1}) = I$$

Discrete Kalman Filter

$$z_k = Z_k - H_k x_k^{(-)}$$

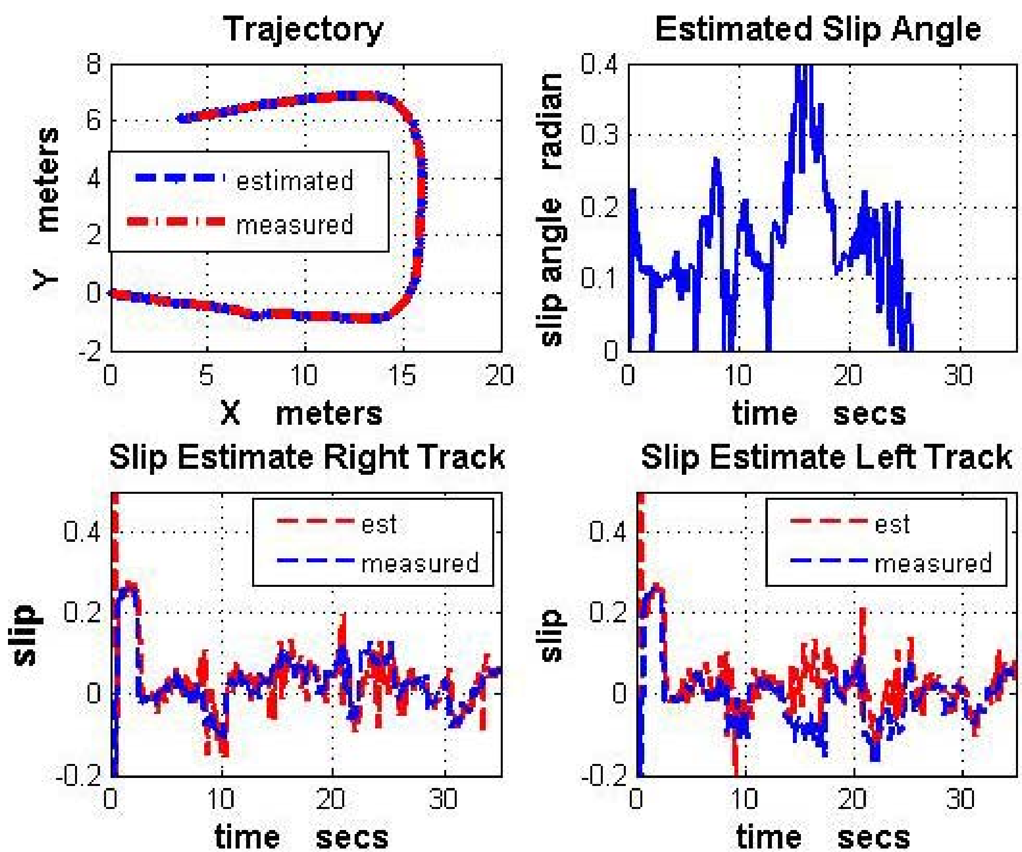


$$x_k^{(+)} = \hat{x}_k^{(-)} + K_k z_k$$

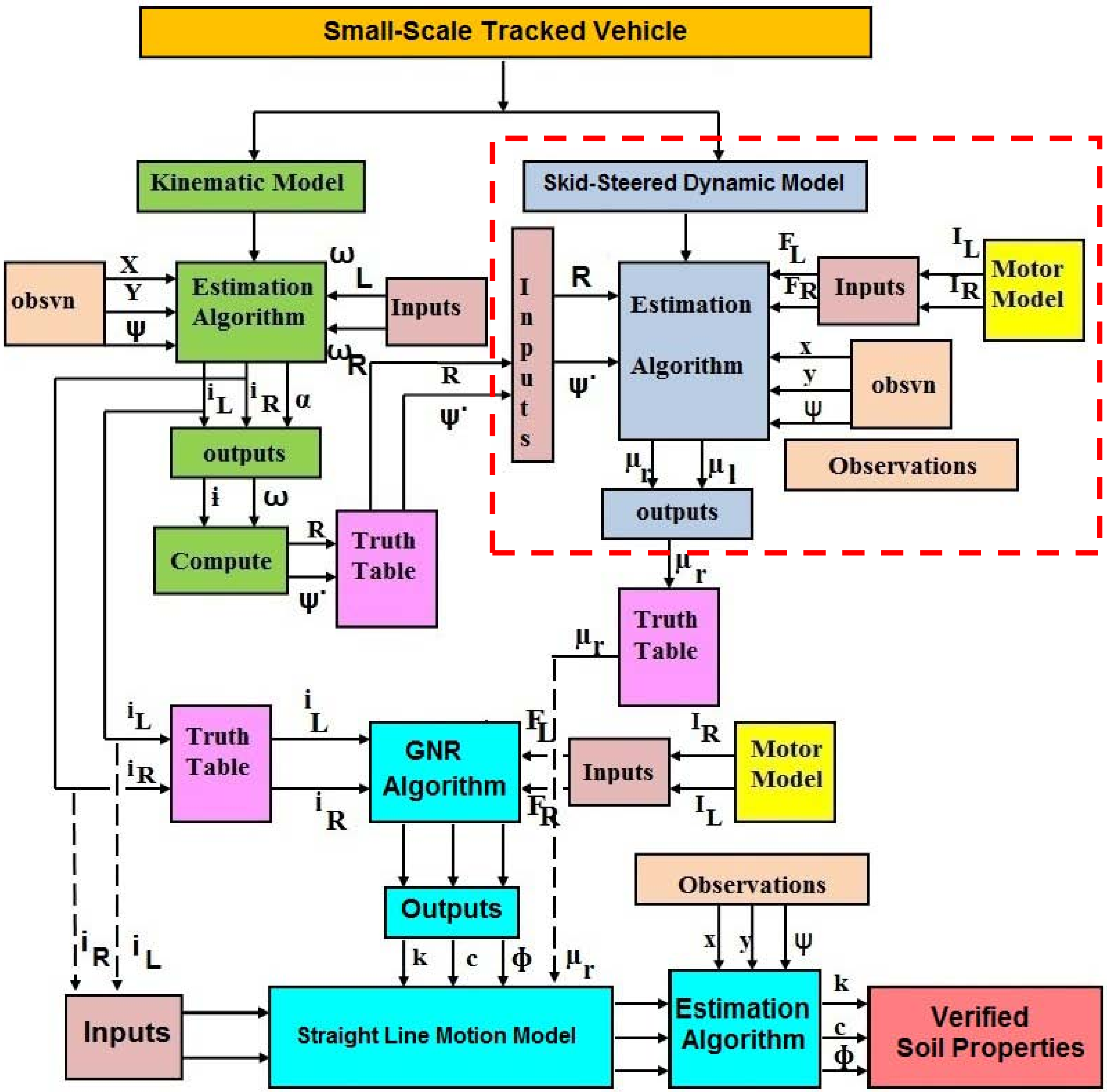
Kalman Gain

$$x(k) = \begin{bmatrix} X(k) \\ Y(k) \\ \Psi(k) \\ i_L(k) \\ i_R(k) \\ \alpha(k) \end{bmatrix}$$

Slip Estimation – Sand Court



Algorithm – 2



α = slip angle

c = soil cohesion

F_L = tractive force left track

F_R = tractive force right track

i = slip on left/right track

i_L = slip on left track

i_R = slip on right track

k = deformation parameter

obsvn = observations

ϕ = soil friction angle

ψ = yaw angle

$\dot{\psi}$ = yaw rate/turning rate

I_L = current left motor

I_R = current right motor

ω_L = angular speed left sprocket

ω_R = angular speed right sprocket

μ_l = lateral coefficient of friction

μ_r = longitudinal coefficient of friction

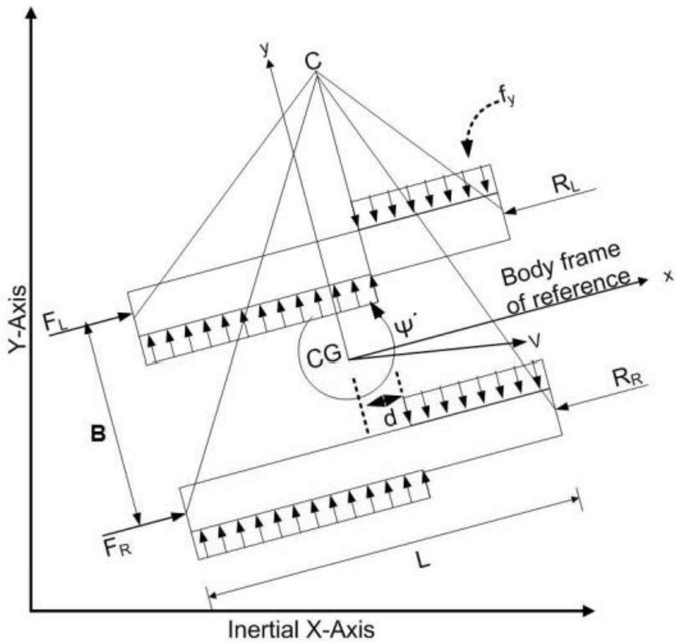
x = displacement in x-direction body fixed coordinates

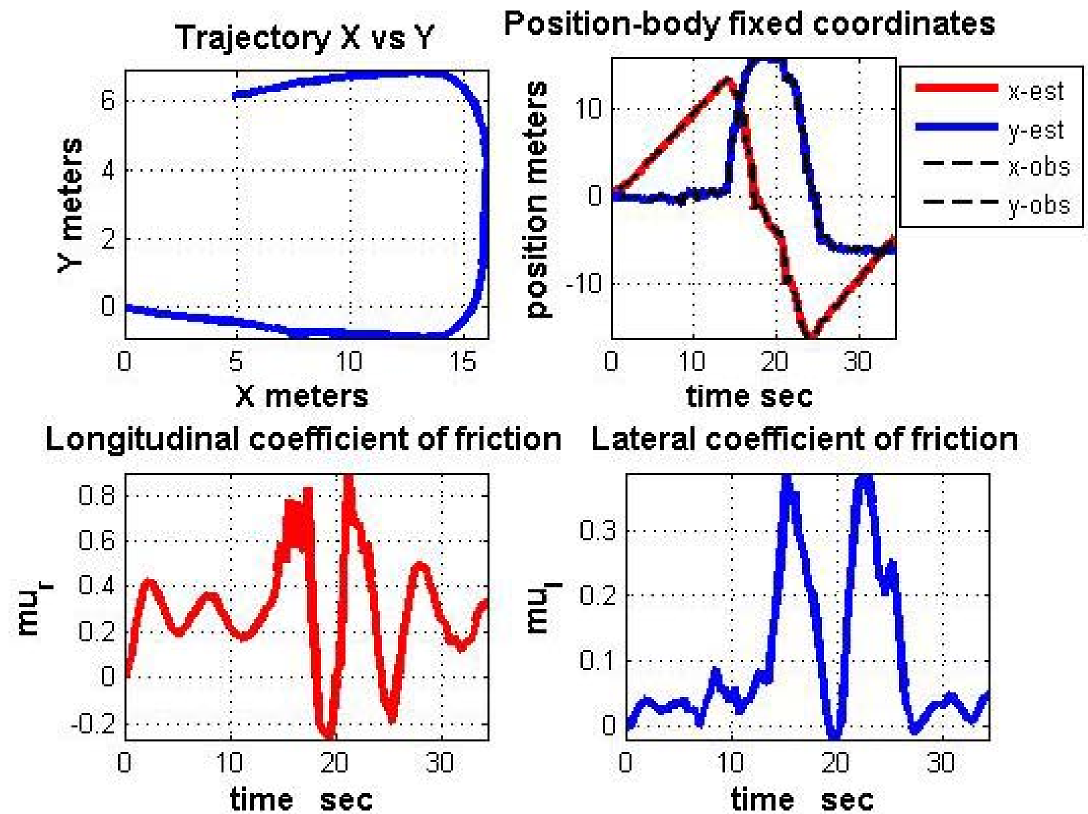
y = displacement in y-direction body fixed coordinates

X = displacement in X-direction inertial coordinate

Y = displacement in Y-direction inertial coordinate

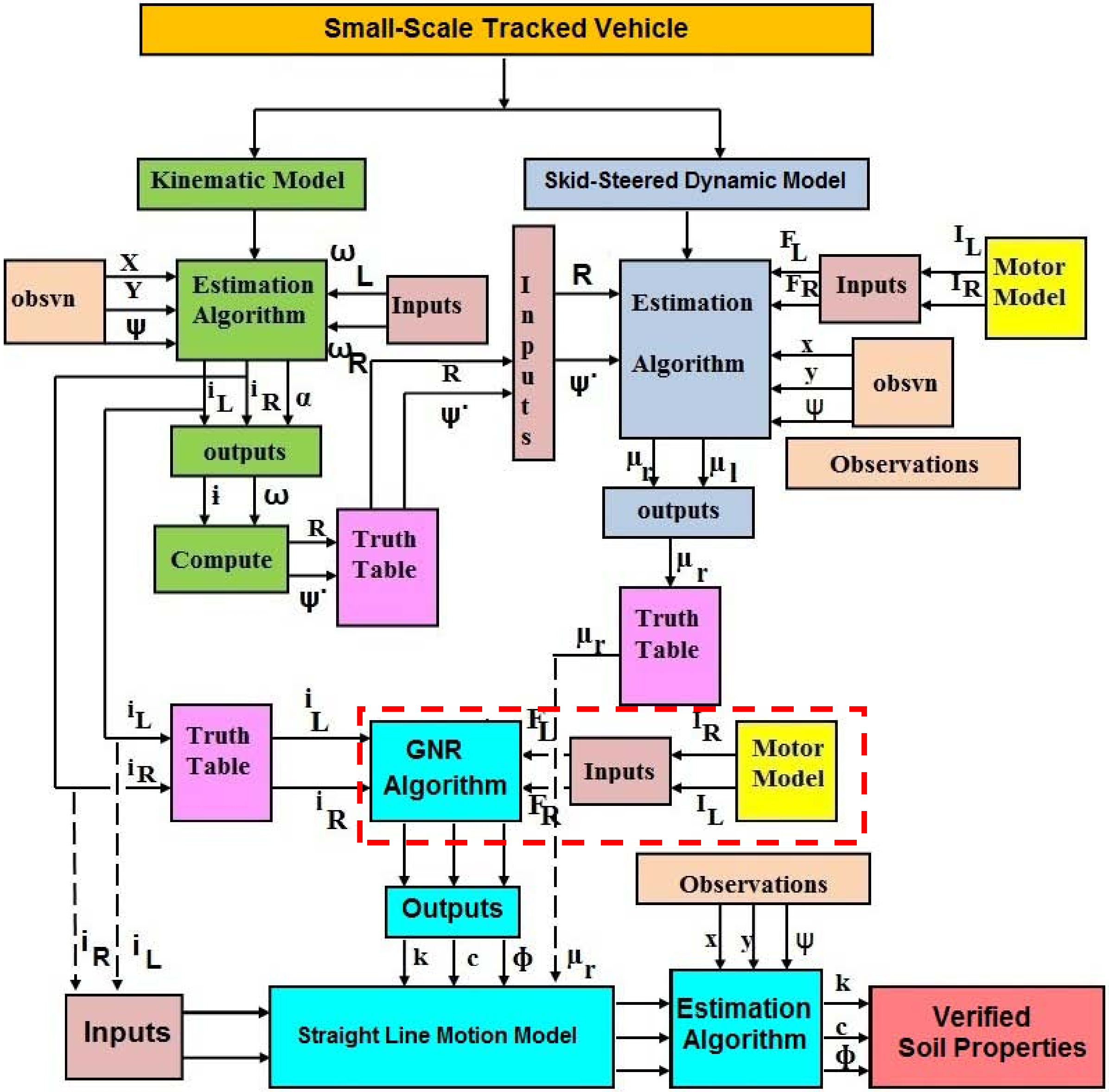
Coefficients of Friction





Note: Probably should conduct some comparisons to existing data sets to see how well these coefficients predict trajectories.

Algorithm – 3



α = slip angle

c = soil cohesion

F_L = tractive force left track

F_R = tractive force right track

i = slip on left/right track

i_L = slip on left track

i_R = slip on right track

k = deformation parameter

obsvn = observations

ϕ = soil friction angle

ψ = yaw angle

$\dot{\psi}$ = yaw rate/turning rate

I_L = current left motor

I_R = current right motor

ω_L = angular speed left sprocket

ω_R = angular speed right sprocket

μ_l = lateral coefficient of friction

μ_r = longitudinal coefficient of friction

x = displacement in x-direction body fixed coordinates

y = displacement in y-direction body fixed coordinates

X = displacement in X-direction inertial coordinate

Y = displacement in Y-direction inertial coordinate

Frictional Dry Sand

TABLE 2.3 Terrain Values

Terrain	Moisture Content (%)	n	k_c lb/in.$^{n+1}$	k_c kN/m^{n+1}	k_ϕ lb/in.$^{n+2}$	k_ϕ kN/m^{n+2}	c lb/in.^{2}	c kPa	ϕ deg
Dry sand (Land Locomotion Lab., LLL)	0	1.1	0.1	0.99	3.9	1528.43	0.15	1.04	28°
Sandy loam (LLL)	15	0.7	2.3	5.27	16.8	1515.04	0.25	1.72	29°
	22	0.2	7	2.56	3	43.12	0.2	1.38	38°
Sandy loam Michigan (Strong,	11	0.9	11	52.53	6	1127.97	0.7	4.83	20°
	23	0.4	15	11.42	27	808.96	1.4	9.65	35°

From J.Y. Wong, Theory of Ground Vehicles, Wiley-Interscience, 3rd ed.

Data	K, m	Cohesion, kpa	Phi, degree	Fmax, N
Obsvn 1	0.01197	1210	28.361	84.115
Obsvn 2	0.00925	1350	25.386	73.944
Obsvn 3	0.01627	1085	30.913	93.3
Obsvn 4	0.022	1350	36.91	117.04
Obsvn 5	0.009	1530	23.03	66.25

Frictional Dry Sand

TABLE 2.3 Terrain Values

Terrain	Moisture Content (%)	n	k_c lb/in.$^{n+1}$	k_c kN/m^{n+1}	k_ϕ lb/in.$^{n+2}$	k_ϕ kN/m^{n+2}	c lb/in.^{2}	c kPa	ϕ deg
Dry sand (Land Locomotion Lab., LLL)	0	1.1	0.1	0.99	3.9	1528.43	0.15	1.04	28°
Sandy loam	15	0.7	2.3	5.27	16.8	1515.04	0.25	1.72	29°
(LLL)	22	0.2	7	2.56	3	43.12	0.2	1.38	38°
Sandy loam	11	0.9	11	52.53	6	1127.97	0.7	4.83	20°
Michigan (Strong, Buchele)	23	0.4	15	11.42	27	808.96	1.4	9.65	35°
Sandy loam	26	0.3	5.3	2.79	6.8	141.11	2.0	13.79	22°
(Hanamoto)	32	0.5	0.7	0.77	1.2	51.91	0.75	5.17	11°
Clayey soil	38	0.5	12	13.19	16	692.15	0.6	4.14	13°
(Thailand)	55	0.7	7	16.03	14	1262.53	0.3	2.07	10°
Heavy clay	25	0.13	45	12.70	140	1555.95	10	68.95	34°
(Waterways Experiment Stn., WES)	40	0.11	7	1.84	10	103.27	3	20.69	6°
Lean clay	22	0.2	45	16.43	120	1724.69	10	68.95	20°
(WES)	32	0.15	5	1.52	10	119.61	2	13.79	11°
LETE sand (Wong)		0.79	32	102	42.2	5301	0.19	1.3	31.1°
Upland sandy loam (Wong)	51	1.10	7.5	74.6	5.3	2080	0.48	3.3	33.7°
Rubicon sandy loam (Wong)	43	0.66	3.5	6.9	9.7	752	0.54	3.7	29.8°
North Gower clayey loam (Wong)	46	0.73	16.3	41.6	24.5	2471	0.88	6.1	26.6°
Grenville loam (Wong)	24	1.01	0.008	0.06	20.9	5880	0.45	3.1	29.8°
Snow (U.S.)		1.6	0.07	4.37	0.08	196.72	0.15	1.03	19.7°
(Harrison)		1.6	0.04	2.49	0.10	245.90	0.09	0.62	23.2°
Snow (Sweden)		1.44	0.3	10.55	0.05	66.08	0.87	6	20.7°

Source: Reference 2.3, 2.4, 2.35, and 2.36.

From J.Y. Wong, Theory of Ground Vehicles, Wiley-Interscience.

Tables from Gelb

TABLE 4.2-1 SUMMARY OF DISCRETE KALMAN FILTER EQUATIONS

System Model	$\underline{x}_k = \Phi_{k-1}\underline{x}_{k-1} + \underline{w}_{k-1}, \quad \underline{w}_k \sim N(\underline{0}, Q_k)$
Measurement Model	$\underline{z}_k = H_k\underline{x}_k + \underline{v}_k, \quad \underline{v}_k \sim N(\underline{0}, R_k)$
Initial Conditions	$E[\underline{x}(0)] = \hat{\underline{x}}_0,\ E[(\underline{x}(0) - \hat{\underline{x}}_0)(\underline{x}(0) - \hat{\underline{x}}_0)^T] = P_0$
Other Assumptions	$E[\underline{w}_k\underline{v}_j^T] = 0$ for all j, k
State Estimate Extrapolation	$\hat{\underline{x}}_k(-) = \Phi_{k-1}\hat{\underline{x}}_{k-1}(+)$
Error Covariance Extrapolation	$P_k(-) = \Phi_{k-1}P_{k-1}(+)\,\Phi_{k-1}^T + Q_{k-1}$
State Estimate Update	$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k[\underline{z}_k - H_k\hat{\underline{x}}_k(-)]$
Error Covariance Update	$P_k(+) = [I - K_kH_k]\,P_k(-)$
Kalman Gain Matrix	$K_k = P_k(-)\,H_k^T[H_kP_k(-)H_k^T + R_k]^{-1}$

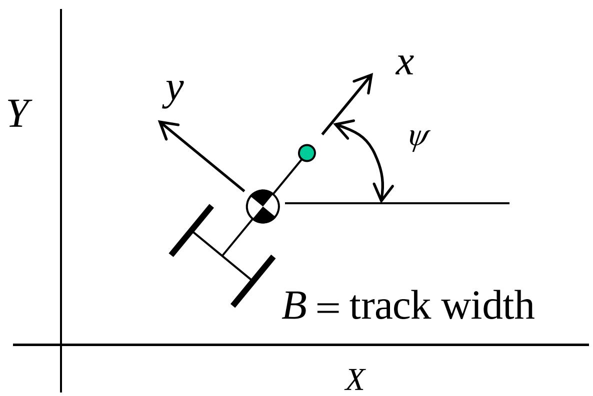
TABLE 6.1-1 SUMMARY OF CONTINUOUS-DISCRETE EXTENDED KALMAN FILTER

System Model	$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t),t) + \underline{w}(t)\,; \quad \underline{w}(t) \sim N(\underline{0}, Q(t))$
Measurement Model	$\underline{z}_k = \underline{h}_k(\underline{x}(t_k)) + \underline{v}_k\,; \quad k = 1, 2, \ldots\,; \quad \underline{v}_k \sim N(\underline{0}, R_k)$
Initial Conditions	$\underline{x}(0) \sim N(\hat{\underline{x}}_0, P_0)$
Other Assumptions	$E\left[\underline{w}(t)\,\underline{v}_k^T\right] = 0$ for all k and all t
State Estimate Propagation	$\dot{\hat{\underline{x}}}(t) = \underline{f}(\hat{\underline{x}}(t),t)$
Error Covariance Propagation	$\dot{P}(t) = F(\hat{\underline{x}}(t),t)\,P(t) + P(t)\,F^T(\hat{\underline{x}}(t),t) + Q(t)$
State Estimate Update	$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k[\underline{z}_k - \underline{h}_k(\hat{\underline{x}}_k(-))]$
Error Covariance Update	$P_k(+) = [I - K_k\,H_k(\hat{\underline{x}}_k(-))]\,P_k(-)$
Gain Matrix	$K_k = P_k(-)\,H_k^T(\hat{\underline{x}}_k(-))\left[H_k(\hat{\underline{x}}_k(-))\,P_k(-)\,H_k^T(\hat{\underline{x}}_k(-)) + R_k\right]^{-1}$
Definitions	$F(\hat{\underline{x}}(t),t) = \left.\dfrac{\partial \underline{f}(\underline{x}(t),t)}{\partial \underline{x}(t)}\right\vert_{\underline{x}(t)=\hat{\underline{x}}(t)}$ $H_k(\hat{\underline{x}}_k(-)) = \left.\dfrac{\partial \underline{h}_k(\underline{x}(t_k))}{\partial \underline{x}(t_k)}\right\vert_{\underline{x}(t_k)=\hat{\underline{x}}_k(-)}$

Ideal Kinematic Steer Models

For a *kinematic* model of a differentially-driven vehicle, we assume there is **no slip**, and that the wheels have controllable speeds, ω_1 and ω_2. The velocity of the CG in the local reference frame has a net effect from each wheel, composed as,

$$\left.\begin{aligned}\dot{x}_1 &= R_w\omega_1\\ \dot{x}_2 &= R_w\omega_2\end{aligned}\right\}\ \text{Note these are the velocities at the wheels.}$$



$$\therefore\ \dot{x} = \tfrac{1}{2}\left(\dot{x}_1 + \dot{x}_2\right) = \tfrac{1}{2}R_w(\omega_1 + \omega_2)$$

The lateral motion is constrained, so, $\dot{y} = 0$

The yaw rate is also composed by the net (constrained) motion of the two wheels, and you can show that:

$$\dot{\psi} = \frac{R_w}{B}\left(\omega_1 - \omega_2\right)$$

So the velocities in the global reference frame are,

$$\dot{\mathbf{q}}_I = \begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\psi}\end{bmatrix} = \mathbf{\Psi}(\psi)\cdot\dot{\mathbf{q}} = \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\dfrac{R_w}{2}\cos\psi(\omega_1+\omega_2)\\ \dfrac{R_w}{2}\sin\psi(\omega_1+\omega_2)\\ \dfrac{R_w}{B}(\omega_1-\omega_2)\end{bmatrix}$$

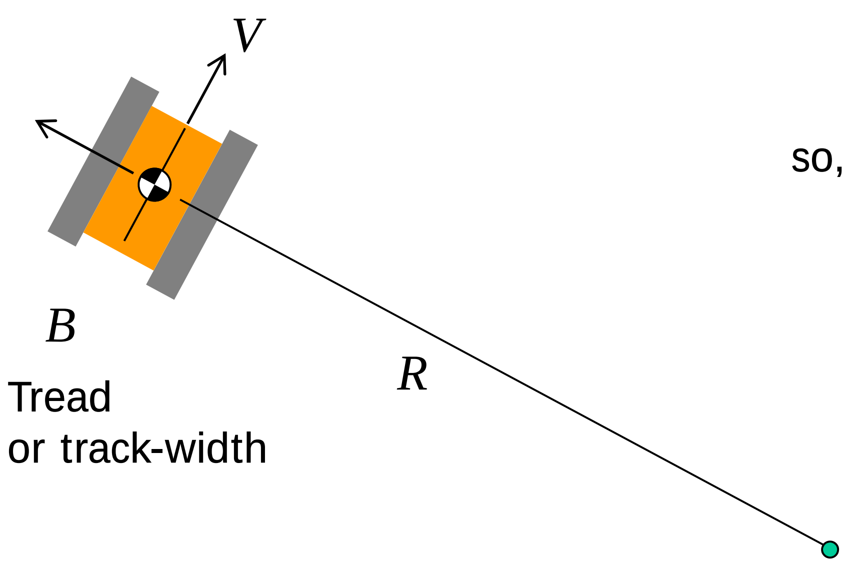
Introduce slip

With slip, the velocity of each track is, $i = \dfrac{r\omega - V}{r\omega} \Rightarrow V = r\omega[1 - i]$

The velocity of the body CG is then,

$$V = \frac{1}{2}(V_R + V_L) = \frac{r}{2}\left[\omega_R(1 - i_R) + \omega_L(1 - i_L)\right]$$

We can estimate the **turning radius** from, $\dfrac{V}{R} = \dot{\psi}$, and the **turning rate**,



$$\dot{\psi} = \frac{r}{2R}\left[\omega_R(1 - i_R) + \omega_L(1 - i_L)\right]$$

so,

$$R = \frac{B}{2}\frac{\left[\omega_R(1 - i_R) + \omega_L(1 - i_L)\right]}{\left[\omega_R(1 - i_R) - \omega_L(1 - i_L)\right]}$$

Kinematic Steer with Slip

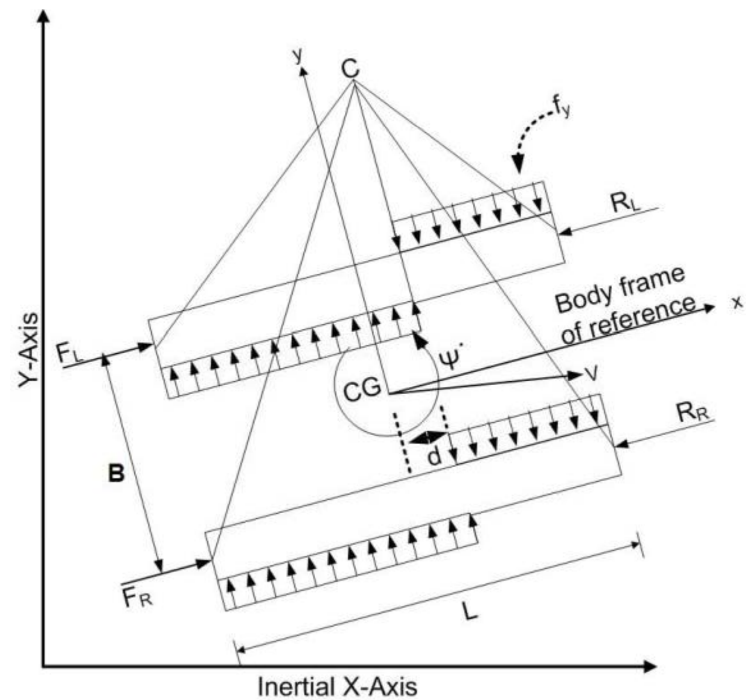
$$\tan\alpha = \frac{-\dot{y}}{\dot{x}}$$

$$i = \frac{r\omega - V}{\max\left(r\omega, V\right)}$$

$$\dot{X} = \frac{r}{2}\left[\omega_L\left(1-i_L\right)+\omega_R\left(1-i_R\right)\right]\left(\cos\psi + \tan\alpha\sin\psi\right)$$

$$\dot{Y} = \frac{r}{2}\left[\omega_L\left(1-i_L\right)+\omega_R\left(1-i_R\right)\right]\left(\sin\psi - \tan\alpha\cos\psi\right)$$

$$\dot{\psi} = \frac{r}{2B}\left[\omega_R\left(1-i_R\right)-\omega_L\left(1-i_L\right)\right]$$



Conclusions

PreMoStat

We can begin understanding how and when the level of uncertainty makes UGV operation impractical.

Closer to having a practical means for quantifying how and when a small UGV is influenced by terrain variability.

The implementation on a mobile stand-alone system may be a reasonable next-step.

It should be possible to incorporate different traction models into this methodology.

Prior information about and/or online estimation of vehicle-terrain interaction parameters can help improve UGV traversability on terrains having significant variability.

Points of Contact

Griffin Technologies

- Eric Krotkov ekrotkov@griffin-technologies.com
- Doug Hackett dhackett@griffin-technologies.com

University of Texas at Austin

- Raul Longoria longoria@mail.utexas.edu

Southwest Research Institute

- Andy Moore andrew.moore@swri.org